

PICTURE SYSTEM 2/MULTI PICTURE SYSTEM

PDP-11

REFERENCE MANUAL

Evans & Sutherland Computer Corporation,
580 Arapeen Drive
Salt Lake City, Utah 84108

First Edition
Second Edition
Third Edition
Fourth Edition

November 22, 1976
November 22, 1977
November 30, 1979
November 10, 1980

All information contained herein together with all drawings, diagrams and specifications herein or attendant hereto are, and remain, the property of Evans & Sutherland Computer Corporation. Many of the intellectual and technical concepts described herein are proprietary to Evans & Sutherland and may be covered by U.S. and Foreign Patents or Patents Pending or are protected as trade secrets. Any dissemination of this information or reproduction of this material for commercial or other purposes other than the express purposes for which it has been made available are strictly forbidden unless prior written permission is obtained from Evans & Sutherland Computer Corporation.

(C)

Copyright Evans & Sutherland Computer Corporation 1976,77,79,80

This manual was prepared by:

M. Mantle
D. Mortenson

All reference to this document should be made to:
Document: E&S #901171-051 A3 (formerly #901130-001)

Evans & Sutherland Computer Corporation assumes no responsibility for any errors that may appear in this manual. The information in this document is subject to change without notice.

PREFACE

The Picture System 2 and Multi Picture System PDP-11 Reference Manual provides the hardware details essential in understanding the Picture System 2, Multi Picture System and the PDP-11 interface at a system level.

Chapter 1 contains an overview of the Picture System 2 and Multi Picture System hardware at a functional level.

Chapter 2 describes the hardware details of Picture System 2, Multi Picture System and the interface to the PDP-11. The information contained in this chapter is detailed to provide a full understanding of the Picture System 2 and Multi Picture System hardware components.

Chapter 3 details the system conventions which should be followed in programming the Picture System 2 and Multi Picture System at the hardware level.

Chapter 4 contains the 4x4 matrix transformations used to implement rotation, translation, scaling, windowing, etc. for the Picture System 2 and Multi Picture System.

Appendix A contains the hardware specifications for the Picture System 2 and Multi Picture System. These specifications state the hardware capabilities for these systems.

Appendix B contains an example program which illustrates the programming of the Picture System 2 and Multi Picture System at the hardware level.

The term Picture System is used to refer to the Picture System 2 and the Multi Picture System throughout this document. Since the basic components and capabilities of these two systems are the same, all references to the Picture System are intended to apply to both systems.

Throughout this manual, three number systems will be used - octal, binary, and decimal. To avoid cluttering all numbers with subscripted bases, the following general convention will be used:

Octal - for address locations, contents of addresses, character codes and operation codes for instructions; in most cases there will be words of 6 octal digits.

Binary - for describing a single binary element.

Decimal - for all normal referencing to quantities.

TABLE OF CONTENTS

CHAPTER 1	OVERVIEW OF THE PS2 AND MPS	
1.1	Picture Controller Interface.....	1-6
1.2	Picture Data Bus.....	1-7
1.3	Picture Processor.....	1-8
1.4	Picture Memory.....	1-10
1.5	Picture Generator - Picture Display.....	1-11
1.5.1	Refresh Controller/ Refresh and Device Processor.....	1-11
1.5.1.1	Refresh Controller.....	1-11
1.5.1.2	Refresh and Device Processor.....	1-12
1.5.2	Character Generator.....	1-14
1.5.3	Line Generator.....	1-15
1.5.4	Picture Display.....	1-17
1.6	Picture System Interactive Devices.....	1-18
1.6.1	Tablet.....	1-19
1.6.2	Control Dials.....	1-19
1.6.3	Joystick.....	1-19
1.6.4	Function Switches & Lights.....	1-20
1.6.5	Lighted Function Buttons.....	1-20
1.6.6	Alphanumeric Keyboard.....	1-20
1.6.7	Light Pen.....	1-20
CHAPTER 2	PS2 AND MPS HARDWARE DETAILS	
2.1	The Picture Data Bus (PSBUS).....	2-5
2.1.1	The Picture Memory (PSMEM).....	2-6
2.1.2	System Control Block (SCB).....	2-9
2.1.3	Picture System Data Transfers.....	2-10
2.2	Picture System/PDP-11 I/O Interface.....	2-11
2.2.1	Direct I/O Registers.....	2-12
2.2.2	I/O Status Register (IOST): 767670.....	2-14
2.2.3	Direct Memory Access Registers.....	2-18
2.3	The Picture Processor.....	2-20
2.3.1	The MAP Input Controller.....	2-29
2.3.2	The Matrix Arithmetic Processor (MAP).....	2-31
2.3.2.1	MAP Internal Registers.....	2-32

2.3.2.2	Picture Processor Commands.....	2-38
	HALT.....	2-39
	NO-OP.....	2-39
	JUMP.....	2-40
	PUSHJ.....	2-41
	POPJ.....	2-42
	LOAD.....	2-43
	STORE.....	2-44
	LOADSTK.....	2-45
	STORESTK.....	2-46
	XFER.....	2-47
	PUSH.....	2-48
	POP.....	2-49
	MATCON.....	2-50
	2DDRAW.....	2-53
	3DDRAW.....	2-53
	4DDRAW.....	2-53
2.3.3	The MAP Output Formatter.....	2-57
2.3.4	Picture Processor Maintenance Registers.....	2-67
2.4	The Picture Generator.....	2-75
2.4.1	The Line Generator Input Controller.....	2-76
2.4.2	The Refresh Controller/ Refresh and Device Processor.....	2-76
2.4.2.1	The Refresh Controller.....	2-77
2.4.2.2	The Refresh and Device Processor.....	2-85
2.4.2.2.1	Refresh and Device Processor Data Structures.....	2-96
2.4.2.2.2	Refresh and Device Processor Commands.....	2-111
	NOP COMMAND (0).....	2-112
	REFRESH COMMAND (1).....	2-113
	INITIALIZATION COMMAND (2).....	2-114
	ADD COMMAND (3).....	2-115
	REPLACE COMMAND (4).....	2-119
	COMPACT COMMAND (5).....	2-123
	DELETE/COMPACT COMMAND (7,10).....	2-127
	REPLACE/COMPACT COMMAND (11,12).....	2-131
	CHANGE STATUS COMMAND (13).....	2-135
	CURSOR COMMAND (14).....	2-137
	LIGHT PEN COMMAND (15).....	2-142
	DEVICE CONTROL COMMANDS (16,17,20).....	2-147
2.4.2.2.3	Refresh and Device Processor Maintenance.....	2-157
2.4.3	The Line Generator.....	2-159
2.4.4	The Character Generator.....	2-178
2.4.4.1	Standard Character Sizes and Definitions.....	2-179
2.4.4.2	The Character Control.....	2-185
2.4.4.3	The Character Memory.....	2-186
2.4.4.4	The Character Multiplier.....	2-192
2.4.4.5	Detailed Programming of the Character Generator...	2-197
2.4.5	Picture Generator Maintenance Registers.....	2-205

2.5	Interrupt Control.....	2-215
2.5.1	Picture System 2 Interrupt Control.....	2-216
2.5.1.1	PS2 Real-Time Clock Interrupt Control.....	2-216
2.5.1.2	System Interrupt Control.....	2-218
2.5.1.3	Device Interrupt Control.....	2-222
2.5.2	Multi Picture System Interrupt Control.....	2-225
2.6	Picture System Devices.....	2-229
2.6.1	Real-Time Clock.....	2-230
2.6.2	Data Tablet.....	2-233
2.6.3	Alphanumeric Keyboard.....	2-237
2.6.4	Function Switches & Lights/ Lighted Function Buttons.....	2-238
2.6.5	Control Dials/Joystick.....	2-240
2.6.6	Light Pen.....	2-241
2.6.7	Remote Terminal Interface.....	2-247
2.6.8	User Devices.....	2-258
2.7	Map of the System Control Block (SCB).....	2-260

CHAPTER 3 PICTURE SYSTEM SOFTWARE CONVENTIONS

CHAPTER 4 THE 4x4 MATRIX TRANSFORMATIONS USED BY THE PICTURE SYSTEM

4.1	The PS2 Graphics Software Package.....	4-2
	GETROT.....	4-3
	GETSCL.....	4-4
	GETTRN.....	4-5
	HITWIN.....	4-6
	INST.....	4-7
	MASTER.....	4-8
	PSINIT.....	4-9
	ROT.....	4-10
	SCALE.....	4-11
	TRAN.....	4-12
	WINDOW.....	4-13
4.2	The MPS Graphics Software Package.....	4-14
	HBEGIN.....	4-15
	MPINIT.....	4-16
	TIDENT.....	4-17
	TINST.....	4-18
	TMAST.....	4-19
	TROTX.....	4-20
	TROTY.....	4-21
	TROTZ.....	4-22
	TSCALE.....	4-23
	TTRAN.....	4-24
	TWIND.....	4-25
	TWINDP.....	4-26

APPENDIX A PICTURE SYSTEM SPECIFICATIONS

APPENDIX B PROGRAMMING THE PICTURE SYSTEM

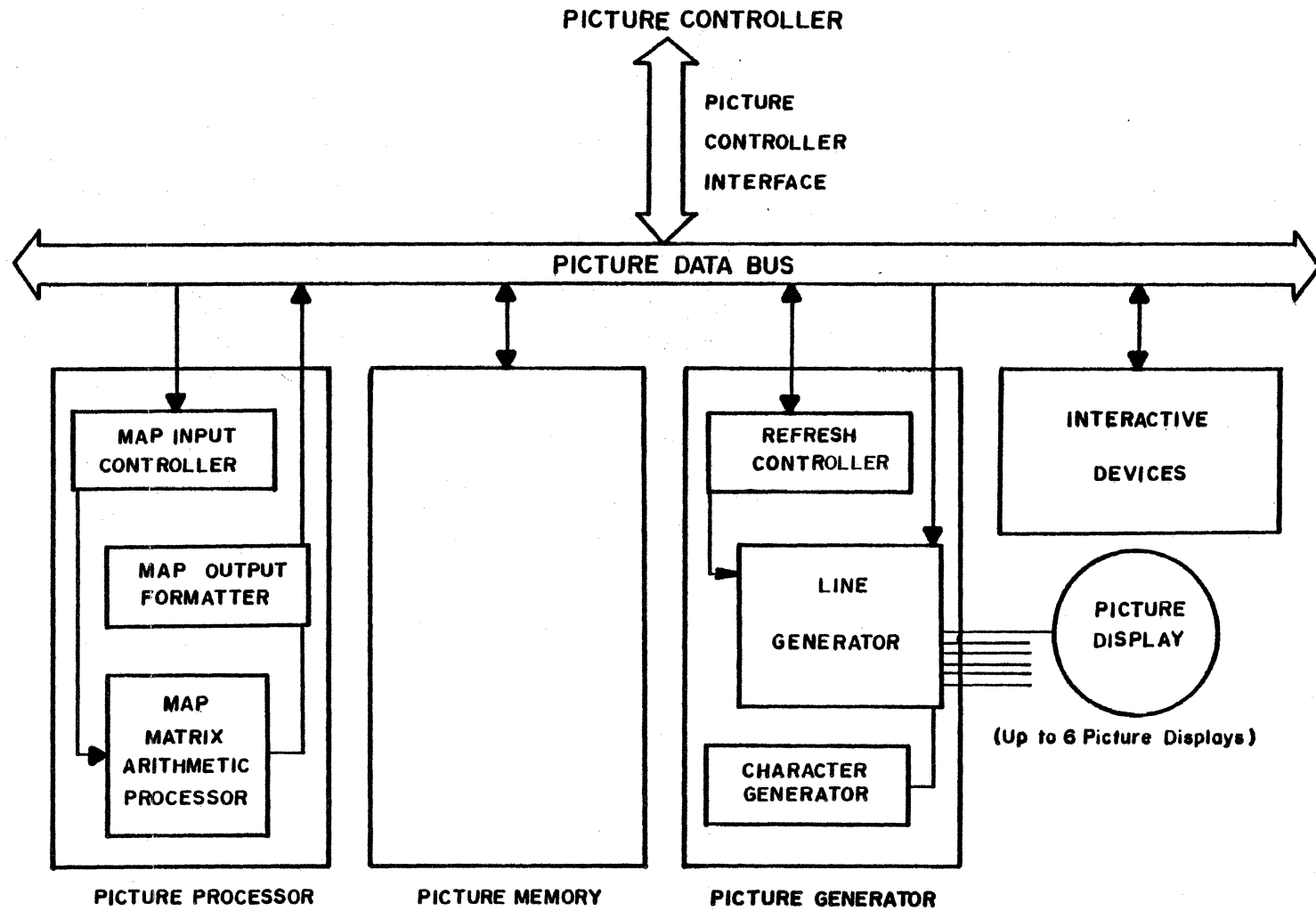
B.1	Introduction.....	B-1
B.2	Program Description.....	B-2
B.3	MACRO-11 Picture System 2 Program Example.....	B-4

CHAPTER 1

1. OVERVIEW OF THE PICTURE SYSTEM 2 AND MULTI PICTURE SYSTEM

This chapter provides an overview of the hardware components of the Picture System 2 and Multi Picture System. The basic hardware components comprising these two systems are the Picture Controller Interface, Picture Data Bus, Picture Processor, Picture Memory, Picture Generator, Picture Display(s) and Interactive Devices. These hardware components are common between the two systems with the primary difference being the replacement of the unit of the Picture Generator that controls refreshing (the "Refresh Controller") in the Picture System 2 with a more intelligent "Refresh and Device Processor" in the Multi Picture System. The Refresh and Device Processor allows the other components which are dedicated to a single user in the Picture System 2 to be shared among several users for the Multi Picture System without undue software overhead. The processing flow and display of data within the two systems is the same, with the system software provided with the Multi Picture System performing the allocation and distribution of system resources that are shared.

Functional block diagrams of the Picture System 2 and the Multi Picture System are shown in Figures 1-1 and 1-2.



1-2

Figure 1-1

Picture System 2 Components

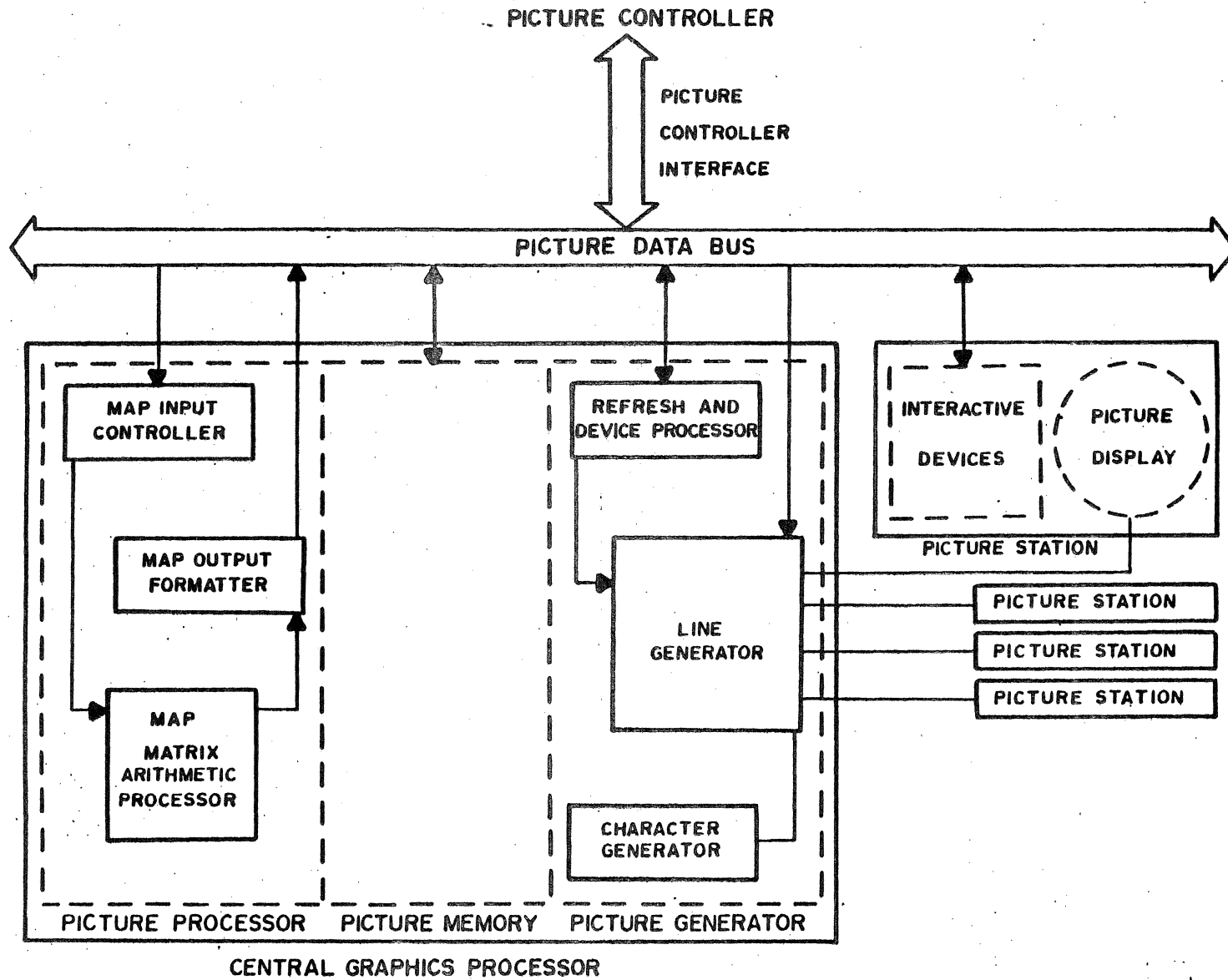


Figure 1-2

Multi Picture System Components

A comparison of Figures 1-1 and 1-2 shows that the Multi Picture System has further categorized the hardware components of the Picture System 2 as a Central Graphics Processor (CGP) and Picture Stations. The Central Graphics Processor includes those components of the Picture System that must be shared among the Picture Stations (i.e., Picture Controller Interface, Picture Processor and Picture Generator). A Picture Station contains all of the resources that are allocated to an individual user of the system and typically includes a Picture System work table, Picture Display and an installation dependent set of interactive devices (i.e., Data Tablet, Control Dials, Keyboard, etc.).

As shown in Figures 1-1 and 1-2, a Picture Controller is required for both the Picture System 2 and the Multi Picture System. The Picture Controller is a general-purpose computer, typically a minicomputer such as Digital Equipment Corporation's PDP-11, which directs the display of pictures on the Picture System. The Picture Controller communicates with the Picture System through a Direct I/O and Direct Memory Access component called the Picture Controller Interface. The Picture Controller Interface allows the Picture System cabinet to be positioned up to 500 feet in distance from the Picture Controller.

The Picture Controller is used to:

- * Contain the data base describing the object(s) to be viewed.
- * Control processing of the object coordinate data by the Picture Processor.
- * Perform all input and output required to facilitate graphical interaction.
- * Compute parameters for use in simulation of object movement, data representation, etc.
- * Perform all standard operating functions required by the operating system under which the control program executes.

PS2/MPS Reference Manual

Overview

All Picture System components and interactive devices are connected to and interact with the synchronous Picture Data Bus. These devices and components, as listed below, are described in detail in the following sections.

- * The Picture Controller Interface
- * The Picture Data Bus
- * The Picture Processor
- * The Picture Memory
- * The Picture Generator
- * The Interactive Devices

1.1 Picture Controller Interface

The Picture Controller Interface has a double cable which runs from the Picture Controller to the Picture System cabinet. This cable allows the Picture System hardware to be located up to 500 feet from the Picture Controller.

Communication between the Picture Controller and the Picture System is achieved through the use of two I/O paths:

1. Direct I/O (DIO) path
2. Direct Memory Access (DMA) path

The Direct I/O path is typically used to pass single-word commands to or from the Picture System. This provides convenient access to the Picture System registers by the computer.

Using the DMA path, blocks of coordinate data may be transferred to or from the Picture System without direct supervision by the Picture Controller.

An example of the use of the Direct I/O path would be the transfer of coordinate data to the Picture System for processing while the DMA path is concurrently transferring the processed data back to the Picture Controller. Another example would be the use of the DMA to transfer data to the Picture Processor for processing at the same time the Direct I/O path is reading the current status of an interactive device. In both cases, these data paths may be used concurrently for information flow.

The Picture System Interface also provides interrupt capabilities for management of Picture System components and interactive devices. The Picture System Interface distributes data to and retrieves data from the various components and devices of the Picture System via the Picture Data Bus.

1.2 Picture Data Bus

All Picture System components and devices connect to and interact with each other on a single, high-speed synchronous data bus. Memory, device registers, and command, control and status registers all exist and are addressable memory locations on the Picture Data Bus.

Use of the Data Bus allows coordinate data to be transferred from the Picture Controller to the Picture Processor while data are concurrently being transferred from the Picture Processor to the Picture Memory. During this process, data are also being transferred from the Picture Memory to the Picture Generator for display. In addition, data may be entered from the Data Tablet or other interactive devices and read by the control program in the Picture Controller.

Data flow is supervised by a bus arbitration system which is integral to the Picture Data Bus.

1.3 Picture Processor

The Picture Processor is a high-speed, microprogrammed, digital arithmetic processor capable of accepting two-, three-, or four-dimensional (homogeneous) data; transforming the data; clipping the transformed data at the boundaries of a six-sided window; performing a perspective division and viewport mapping on the clipped data and outputting the transformed, clipped and viewport mapped data for subsequent display. There are three basic units in the Picture Processor:

- * MAP Input Controller
- * Matrix Arithmetic Processor (MAP)
- * MAP Output Formatter

The MAP Input Controller receives data, typically from the Picture Controller, and channels the data to the Matrix Arithmetic Processor. The MAP Output Formatter receives the processed data from the Matrix Arithmetic Processor and outputs the data format for subsequent display. The MAP Output Formatter may alternately output the data in full 16-bit precision for use by the Picture Controller.

The Matrix Arithmetic Processor (MAP) is the major unit of the Picture Processor and consists of a Transformation Matrix, Transformation Matrix Stack, Parameter Register File and Arithmetic Unit.

The Transformation Matrix is a 4x4 element matrix, where each element is a 16-bit word. This 4x4 matrix is used to transform object coordinate data. It can also be concatenated with other 4x4 matrices to obtain a combined transformation.

The Transformation Matrix Stack is a storage area where up to fourteen 4x4 matrices may be "stacked" or saved for future recall.

The Parameter Register File is an array of 16-bit registers into which parameters specifying viewport boundaries, scale factors, etc. are stored and may be retrieved.

The Arithmetic Unit performs all scalar, vector and matrix arithmetic operations in the Picture Processor. This includes subtraction, addition, multiplication, division and normalization.

The MAP utilizes its units to perform digital operations on the data received from the Picture Controller. These operations are:

- * To transform two-, three- and four-dimensional data.
- * To push the Transformation Matrix onto the Matrix Stack.
- * To pop the top 4x4 matrix of the Matrix Stack into the Transformation Matrix.
- * To load the Transformation Matrix with data from the Picture Controller or Picture Memory.
- * To store the contents of the Transformation Matrix into the Picture Controller or Picture Memory.
- * To concatenate the contents of the Transformation Matrix with a 4x4 matrix in the Picture Controller or Picture Memory to obtain a compound transformation.
- * To load and store the registers of the Picture Processor.
- * To check transformed coordinate data for visibility by comparing with a two- or three-dimensional viewing window. Lines or portions of lines outside the window are removed by a clipping process so that only visible line segments are processed further.
- * Three-dimensional data are converted to two dimensions by computing perspective or orthographic views.
- * To perform a linear mapping of points from the object's coordinate system into that of the Picture Display or any other coordinate system specified.

1.4 Picture Memory

The Picture Memory is a dual-port MOS memory (distinct from the Picture Controller's) organized as addressable 16-bit words. This memory is available in increments of 16K words, expandable to 64K words of memory, dependent upon user requirements.

Extended Memory Multi Picture System Memory is available in increments of 32K words. This memory uses a 22 bit MOS memory where 16 bits are used for data and 6 bits are used for error detection and correction. All single bit errors are detected and corrected and double bit errors are detected. Each picture station may use up to 63.5K words for refresh memory. Any remaining memory may be used for data storage.

Picture Memory may be used in a variety of ways to satisfy the user's application. Typically, a portion of the Picture Memory serves as a refresh buffer into which data, still in digital form, are deposited. This data represents information to be shown on the Picture Display. For each frame displayed, the data are read from the Picture Memory and channeled to the Line Generator where the data are then converted to analog signals to drive the Picture Display.

The ways in which the Picture Memory may be used as a refresh buffer include the single-buffer mode, double-buffer mode and segmented-buffer mode (see Section 1.5.1 for further information).

Multi Picture Systems with more than one Picture Station have Picture Memory divided into sections, with one section for each Picture Station. Each Picture Station display file is then restricted to the size of its section. Subdivision of Picture Memory into sections, which may be of equal or unequal size, is accomplished during software installation and may not be altered during program execution.

1.5 Picture Generator and Picture Display

The Picture Generator is a major component of the Picture System whose primary function is to generate images to be viewed on the Picture Display.

The Picture Generator receives digital data information and converts this data into analog signals which are used to draw the image on the Picture Display.

The Picture Generator consists of three basic units:

- * Refresh Controller/Refresh and Device Processor
- * Character Generator
- * Line Generator

1.5.1 Refresh Controller/Refresh and Device Processor

The Refresh Controller and Refresh and Device Processor perform the same basic functions within the Picture System 2 and Multi Picture System. Each controls the refreshing of images by reading data from the Picture Memory and channeling this data to the Line Generator for display. The only hardware difference between the Picture System 2 and the Multi Picture System is that a Picture System 2 contains a Refresh Controller and the Multi Picture System contains a Refresh and Device Processor. These devices are "plug compatible" on the Picture Data Bus but are not software compatible devices. The Refresh Controller was designed to support a single user of the Picture Generator whereas the Refresh and Device Processor was designed to allow the sharing of the Picture Generator by several users. The addition of the Refresh and Device Processor functionality enables the Multi Picture System to support multiple users. Each of these units are detailed below.

1.5.1.1 Refresh Controller

For the Picture System 2, the Refresh Controller is the unit of the Picture Generator that controls the refreshing of images on the Picture Display. The Refresh Controller reads data from the Picture Memory, a refresh buffer, and channels this data to the Line Generator for display. Under program

supervision, the Refresh Controller is used to manage the organization of the Picture Memory. It also contains special-purpose hardware to facilitate memory segmentation and management.

In single-buffer mode, the entire refresh buffer is used to store a single display frame. In this mode, refresh may be initiated from a partially-updated display frame consisting of some lines from the new frame and some lines from the previous frame.

In double-buffer mode, one-half of the refresh buffer is designated as an old frame and one-half a new frame. Refresh is then initiated from the old frame while the new frame is being constructed. When construction of the new frame is completed, the frame buffers are swapped and the newly-constructed frame is displayed. The space occupied by the old frame becomes available for new frame construction.

The segmented-buffer mode provides the most general use of the refresh buffer for the display and updating of data. Typically, a frame consists of portions which need not be updated as frequently as others. Ideally, these portions should be updated as separate parts, or segments, of the frame. The Refresh Controller facilitates the use of the refresh buffer in this mode by allowing each of the separate portions of the refresh buffer to be given a name by which the segment may be replaced, appended to, deleted, etc.

The Refresh Controller also improves the utilization of the refresh buffer by providing, in segmented-buffer mode, for the reclamation of unused portions of the refresh buffer that have been left by deleted segments. This prevents fragmentation of the refresh buffer into small, unuseable areas.

1.5.1.2 Refresh and Device Processor

For the Multi Picture System, the Refresh and Device Processor controls refreshing of the picture and performs all interactive device input and output. Refresh is accomplished by reading the display file stored in Picture Memory and directing line endpoint coordinates found there to the Line Generator for display and character codes found there to the Character Generator, where they are converted to line endpoints and sent to the Line Generator for display. Refresh is begun at every other tick of a 120 cycle/sec clock (100

cycle/sec for 50Hz power). If the previous refresh is not finished when the second tick occurs, then refresh is begun at the first tick after the previous refresh is completed. This yields a refresh rate of 60Hz, dropping automatically to 40Hz, 30Hz etc. as necessary (50Hz, dropping to 33Hz, 25Hz etc. for 50Hz power).

The Refresh and Device Processor interprets the display file as being organized either in double buffer or segmented buffer mode. In double buffer mode, it continually reads and refreshes the half of the display file where a completed frame of the picture resides, while a new frame is being prepared and deposited in the other half. As soon as preparation of the new frame is completed, the roles of the two buffers are exchanged.

In segmented buffer mode, the display file is partitioned into a number of segments which may be independently updated. The Refresh and Device Processor manages segmentation in the following fashion: When the display program indicates that a particular segment is to be updated, the Refresh and Device Processor combines its refresh operation with a reorganization of the display file which puts all available space immediately after the segment to be updated. It then permits the new version of the segment to enter the display file, starting at the beginning of the available space, while continuing to refresh the entire picture including the old version of that segment. During the first refresh following completion of the segment update, the Refresh and Device Processor bypasses the old version of the updated segment, displays the new one, and overwrites the old with the new to leave all available space at the end of the new segment. This procedure avoids fragmentation of available memory space, preserves the order of the segments, avoids unpleasant visual effects, and insures that mode change commands inserted in one segment will not incorrectly apply to following segments.

Double buffering and segmentation can be combined to the extent that any one segment may be double buffered. When applicable, this approach offers efficiency improvements over pure segmentation, because the steps of old segment overwriting are avoided. The single buffer mode available in earlier Picture System models is not available in the Multi Picture system.

When several display programs are running simultaneously, each may choose its own form of display file organization. The Refresh and Device Processor always refreshes all dis-

play files at the same rate.

The second major function of the Refresh and Device Processor is to manage all interactive devices in the system. Between refreshes, it polls each connected device, compares the value read with the value that last caused an interrupt, and interrupts the Picture Controller if a pre-specified threshold is exceeded.

The Refresh and Device Processor can generate a cursor to correspond with a tablet value read, a tracking cross to follow a light pen, or light settings to correspond with switch settings. The display programs, during their initialization phase, specify what devices are to be connected, what output correspondences are to be made, and what types and ranges of input values are to cause interrupts. Thereafter it acts upon inputs only when interrupted and otherwise ignores interactive devices except to change assignments if desired.

1.5.2 Character Generator

The Character Generator is the unit of the Picture Generator that accepts and interprets character codes from the Line Generator, producing strokes which are then channeled back to the Line Generator for display. The microprogrammable Character Generator provides the basic capabilities to interpret the full 128 ASCII character set for generation of the 95 displayable ASCII character subset and for performing character positioning commands (i.e., carriage returns, line feeds, etc.) The Character Generator also allows the user to establish the left and top margins for use with the character positioning codes.

As a standard feature, the Character Generator is capable of creating regular and italicized characters in eight sizes, ranging from .08 cm (.03") to 1.88 cm (.74") in size. It is also capable of producing subscript and superscript characters which are approximately two-thirds the currently selected character size. A unique capability of the Character Generator is its ability to terminate generation of characters when a character string extends beyond the right boundary of the Picture Display. This hardware feature frees the user from concern of character wrap-around on the display.

The Character Generator is capable of being microprogrammed

by the Picture System user to produce alternate character fonts (or special symbols) at varying sizes and orientations. This ability allows characters to be produced at any orientation (i.e., horizontal, 90-degrees counter-clockwise, 45-degrees counter-clockwise, etc.) while maintaining the correct response to the character positioning commands.

The Character Generator has a memory, half of which is pre-programmed to interpret the 128 ASCII characters as described above, and half is available for the user to program various character fonts and special symbols.

1.5.3 Line Generator

The Line Generator receives data detailing coordinate point positions, status information which describes the modes of operation, and character codes that are passed to the Character Generator for interpretation. This information is used to produce the final image seen by the viewer.

Coordinate point positions are specified by Move or Draw commands in the Picture Display coordinate system. Each coordinate point, defined as X, Y and intensity (Z), causes the Line Generator to position the electron beam of the Picture Display to the location specified. If the command is a Move, the beam will then be positioned without intensification. If the Line Generator is currently in Dot mode, the beam will be positioned and then intensified. If the command is a Draw, a line will be drawn from the current beam location, positioned from a previous Move or Draw command, to the coordinate specified. The Line Generator has the capability of varying the intensity of a line between its endpoints. The line drawn will be in the line texture (dashed, etc.) currently selected for the Line Generator.

A status command to the Line Generator specifies the modes in which lines and characters are to be displayed. These modes are:

- * Blink
- * Texture
- * Color
- * Display Select

PS2/MPS Reference Manual

Overview

Setting the Blink mode indicates that all subsequent lines and characters output to the display are to blink until this mode is reset.

The Texture mode is used to indicate the manner in which all subsequent lines and characters are to be drawn by the Line Generator. Line textures available are: solid line mode; long dashed line mode; long-short dashed line mode; long-short-short dashed line mode and short dashed line mode.

For all of the above dashed line textures, the Line Generator ensures that dashes begin and end at the endpoints on the line.

Conversely, the Line Generator may be set to a textured line mode whereby connected line segments may be displayed without concern for individual line endpoints. This textured line mode effectively allows the appearance of a smoother curve by de-emphasizing the individual line segment endpoints.

Color selection is used in conjunction with a color monitor to initiate changes in color for all subsequent lines and characters which are output to the display after this mode is initiated.

In connection with the color mode selection, an intensity selection is also used to control the drawing speed of the Line Generator. This change in drawing speed is utilized with the beam penetration color displays to present an appearance of equal intensity for the various colors available (i.e., red, red-orange, orange, yellow and green).

The Display Select mode is used in multiple Picture Display configurations in choosing to which Picture Display all subsequent lines and characters are to be output. The Display Select capabilities allow one or more images to be viewed on any combination of the six possible Picture Displays.

Character codes are not interpreted by the Line Generator, but are passed to the Character Generator. The Character Generator interprets the character codes, generating individual strokes which are then channeled back to the Line Generator for display.

The Line Generator is capable of displaying lines and characters at one distinct intensity level. The depth-cue feature increases the number of intensity levels which may be displayed from 1 to 64. This feature also allows the intensity of a single line to vary, from bright to dim according to its Z value, imparting the illusion of depth to three-dimensional images.

1.5.4 Picture Display

The Picture Display receives analog signals from the Picture Generator which are used for electron beam positioning and intensity control. The Picture Generator controls beam positioning and the drawing of all vectors, characters and dots on the Picture Display.

The standard Picture Display is monochromatic and can be supplied with any common phosphor. The standard phosphor is P4, which is white, bright, of extremely short persistence (leaving no trail as lines are moved), and has very fine granularity.

The E&S Color Display System is now also available. This display employs a full-color shadow-mask CRT driven calligraphically. Colors may appear in any of 64 hues and 8 saturations, providing a total of 449 different colors. Speed need not be and should not be varied from color to color. Intensity variation, either depth-cued or selected from the 64 levels provided, is available independent of color.

1.6 Picture System Interactive Devices

All Picture System interactive devices are interfaced directly to the Picture Data Bus. Data may be input from the interactive devices by the control program using the Direct I/O path of the Picture Controller Interface. These interactive devices may be used under interrupt control, polled directly by a control program in the Picture Controller or sampled once per refresh by the Refresh and Device Processor. The Refresh and Device Processor can examine the status of each device and if pre-established conditions are met, an interrupt to the Picture Controller issued.

The following interactive devices are available for the Picture System:

1. Data Tablet
2. Control Dials
3. Joystick
4. Function Switches & Lights
5. Lighted Function Buttons
6. Alphanumeric Keyboard
7. Light Pen

These graphic devices provide all the capabilities typically required for graphical interaction with the Picture System. Appropriate use of these interactive devices, along with the dynamic qualities of the Picture System, provide the user with the tools required for a three-dimensional, interactive graphics system.

1.6.1 Data Tablet

The Data Tablet serves as the standard, general-purpose graphic input device for the Picture System. The tablet can be used for positioning or pointing to the picture elements by use of a stylus, or pen, whose x,y coordinates are read by the Picture Controller. A "cursor" may be generated on the Picture Display to indicate the position of the pen on the tablet. For the Multi Picture System, this generation may be accomplished automatically by the Refresh and Device Processor immediately after the tablet position is read, providing instantaneous response, with no need to interrupt the Picture Controller. In conjunction with the Picture Processor, the tablet and pen can perform the interactive functions usually reserved for such graphic input devices as light pens, joysticks and function switches.

1.6.2 Control Dials

Control Dials are analog inputs which permit the user to control size, position and orientation of objects or other functions required by the application.

1.6.3 Joystick

The Joystick provides the user a three-axis analog input device which may be used to control size, position and orientation of objects that are displayed. A joystick is often used as the input device to control the observers position in real-time simulations.

1.6.4 Functions Switches & Lights

Function Switches & Lights provide the user the capability to utilize switches for functions assigned under program control. Each switch has three positions: "locked on", "off" and "momentary on with spring return to off". The lights may be used to indicate function switch setting or to display programmed information to the user. In the Multi Picture System when the lights are to simply reflect switch settings, the Refresh and Device Processor can automatically keep them current without program intervention.

1.6.5 Lighted Function Buttons

Lighted Function Buttons provide the user the capability to utilize buttons for functions assigned under program control. Each button is backed by a light which may be used to indicate the button setting or to display programmed information to the user. In the Multi Picture System when the buttons are to be lighted to simply reflect button settings, the Refresh and Device Processor can automatically keep them current without program intervention.

1.6.6 Alphanumeric Keyboard

The Alphanumeric Keyboard is a standard 61-key, 128-character keyboard used for text or data input to the Picture Controller for graphical interaction or other functions required by the application.

1.6.7 Light Pen

The Light Pen is available only with the Multi Picture System. It is used for pointing to lines, characters, or structures directly on the CRT. When a "hit" occurs, the display program may determine the "name" of the structure hit, the identity of the line hit, the coordinates of the pen when the hit occurred, or any combination of these pieces of information. A "tracking cross" can be generated on the Picture Display to follow the movements of the Light Pen, and this generation can be made to occur automatically under control of the Refresh and Device Processor. To "find" the pen, "screen blast" is momentarily generated when the pen nears the screen. The "blast" is followed by the appearance of the tracking cross. The Light Pen is a very natural and direct pointing device.

CHAPTER 2

2. PICTURE SYSTEM 2 AND MULTI PICTURE SYSTEM HARDWARE DETAILS

The basic hardware components comprising the Picture System 2 and the Multi Picture System are:

1. The Picture Data Bus (PSBUS) and Memory (PSMEM)
2. The Picture System/PDP-11 I/O Interface
3. The Picture Processor
4. The Picture Generator
5. The Picture Display
6. Interactive Devices

Figures 2-1, 2 and 3 show an overview of the interconnection of each of these components within the Picture System 2 and Multi Picture System. The following sections of this chapter detail the functions performed, command and data formats, and command and status register definitions of each of these components. Those portions of the Picture System which are not common between the Picture System 2 and the Multi Picture System are specifically indicated. In particular, Sections 2.4.2, 2.4.3c, 2.5 and 2.6 contain system specific details.

In the diagrams of the registers, the bits marked "RO" are Read Only, "WO", are Write Only and those bits that are unmarked are Read/Write.

Throughout this chapter, repeated reference is made to the terms INIT, PSRESET, and RESET. INIT is the name given the UNIBUS initialization signal issued by the PDP-11 on power-up, execution of the PDP-11 RESET instruction and the pressing of the START switch on the computer console. PSRESET is the name given the PSBUS initialization signal issued on Picture System power-up or the setting of the PSRESET bit in the I/O status register (see Section 2.2.2). Each Picture System component has its own RESET capability.

In each of the individual component descriptions this is referred to as RESET. PSRESET performs the equivalent of the RESET for all of the components of the Picture System. PSRESET is not issued by the PDP-11 INIT signal.

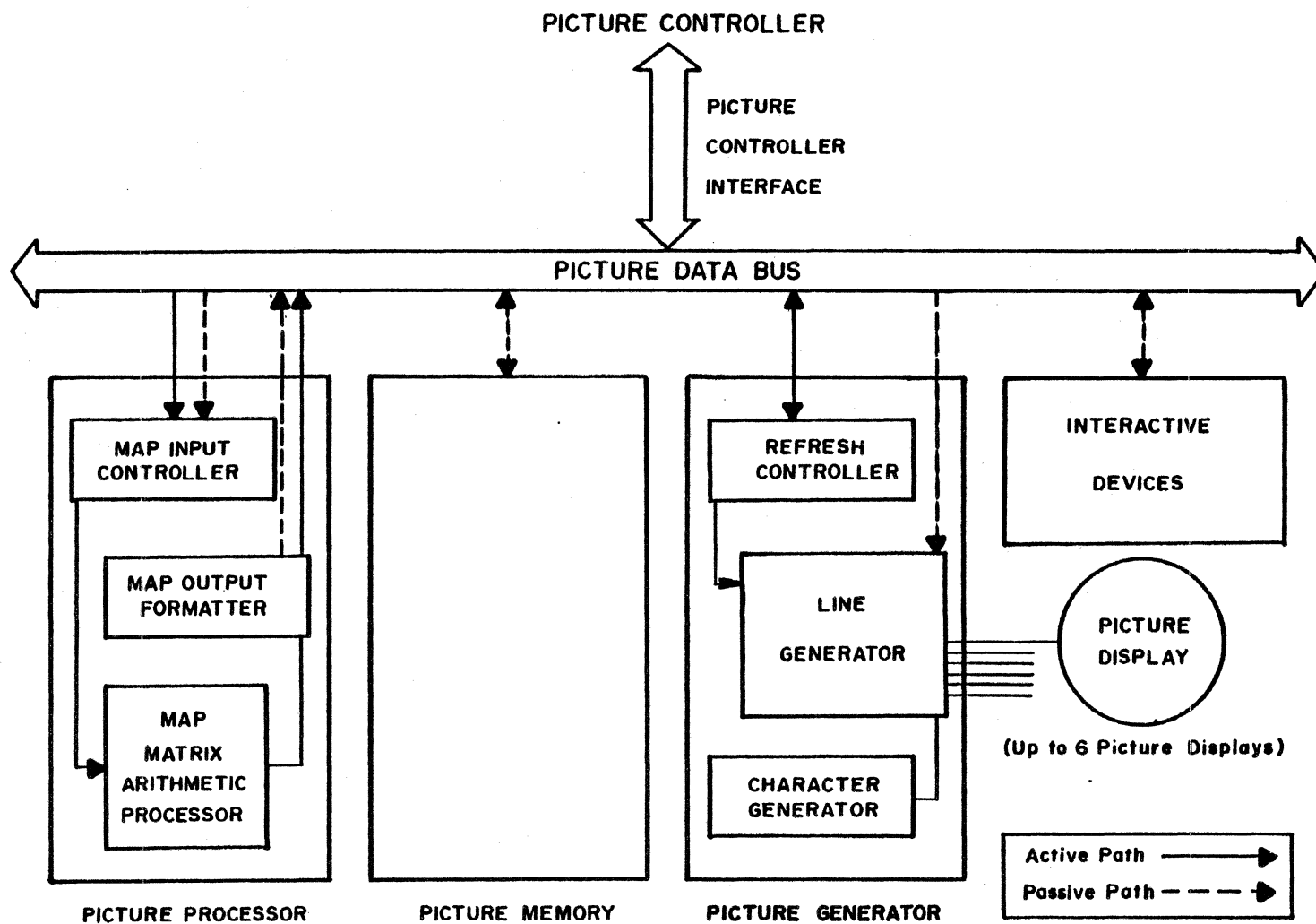
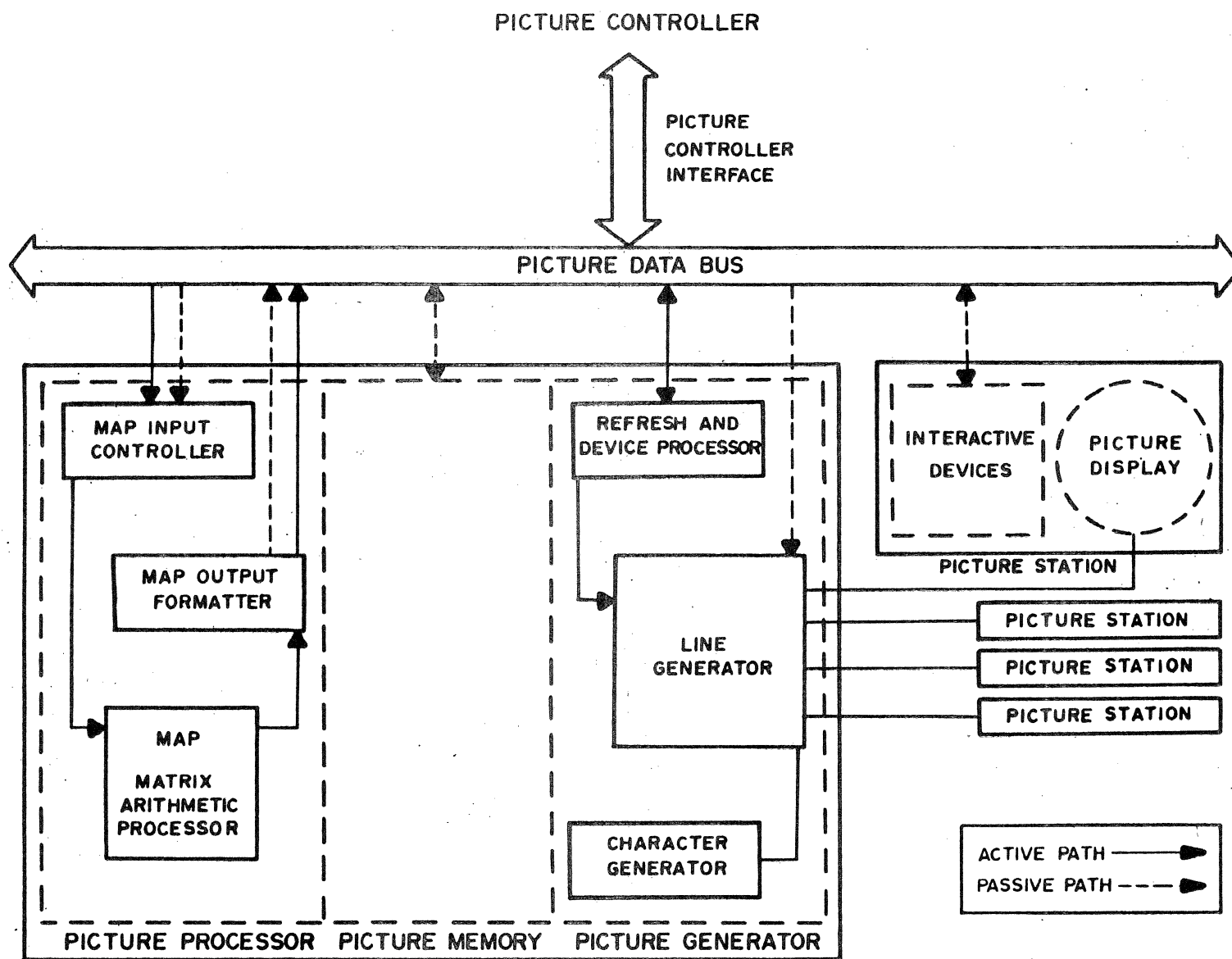


Figure 2-1
Picture System 2 Block Diagram



CENTRAL GRAPHICS PROCESSOR

Figure 2-2

Multi Picture System Block Diagram

2-4

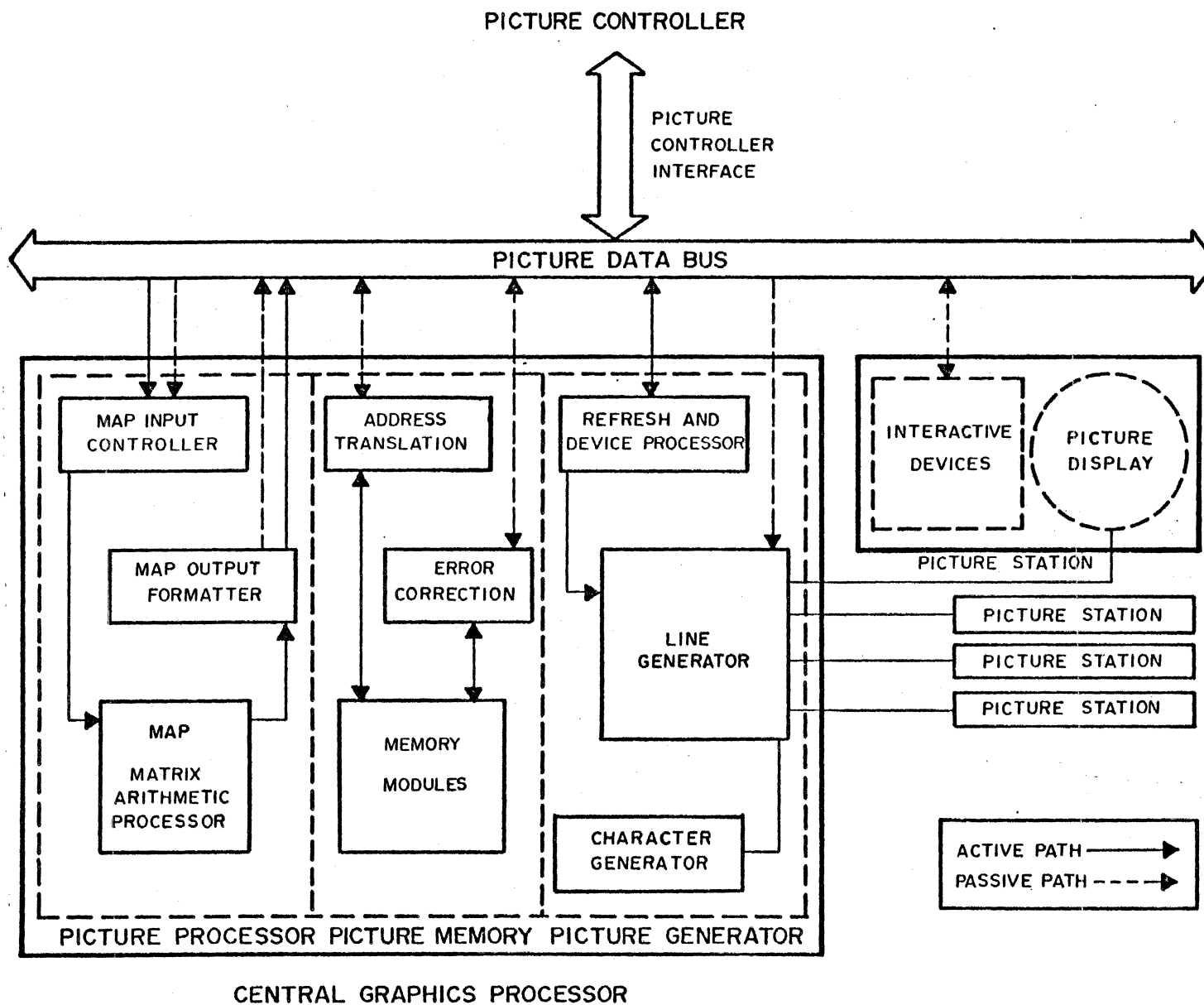


Figure 2-3
Extended Memory Multi Picture System Block Diagram

2.1 The Picture Data Bus (PSBUS)

All Picture System components and devices connect to and communicate with each other on a single, high-speed synchronous data bus (PSBUS). Addresses, data and control information are channeled through the 59 lines of the bus. Memory, device registers, and command, control and status registers are referenced as addressable memory locations on the PSBUS. All devices on the PSBUS (including memory) are considered as either "Active" or "Passive" devices. An Active device can initiate and control the transfer of data on the PSBUS to or from a Passive device. A Passive device will read or write data only when requested to do so by an Active device. An example of an Active device is the Picture System/PDP-11 Direct I/O Interface which actively transfers data to or from the Picture Memory. The Picture Memory, in this example, is the Passive device.

2.1.1 The Picture Memory (PSMEM)

The Picture Memory is a Passive device which consists of a dual-port MOS memory organized as 16-bit words. PSMEM is configured in increments of 16K words of memory dependent upon the system configuration and user requirements.

Certain memory locations have been reserved for system use as command and control registers and device registers. Specifically, the top 256 memory addresses (locations 177400-177777) are reserved. This area is referred to as the System Control Block (SCB). All other memory locations may be used as refresh buffer areas for transformed data that is to be used as refresh buffer areas for transformed data that is to be transformed for subsequent display. Thus, a maximum of 63.75K 16-bit words are available for general program use. It should be noted that the SCB is always located at 177400-177777; hence, actual loss of general purpose memory occurs only when the system has 64K of memory. Additionally, the SCB is always present on the system, including a system with no general-purpose memory, regardless of the actual memory configuration.

The word length in Extended Memory is 22 bits, 16 bits for data and 6 bits for error detection and correction. If a single bit error is detected on a memory read, it is corrected, and the Single Bit Error Status Bit is set in the Extended Memory Status Register. If the Double Bit Error Interrupt Enable bit is enabled an interrupt is generated. Two LED's on the Port Control Board (195442) also indicate the status of the above Error Bits. A green LED monitors the Single Bit Error Status Bit and a red LED monitors the Double Bit Error Status Bit. A LED will be turned on if its respective status bit is set. The above Status bits are cleared by writing a one into the respective bit location.

Each memory access by a Picture System device to Extended Memory is considered a logical memory access that is mapped into a physical memory access. Physical memory is made up of memory pages with each page being 256 memory words in length. The logical memory as seen from the Picture System device is 256 pages (64K words) with the top page (logical page 255) being the System Control Block, and the next to top page (logical page 254) the Picture System Common Memory Block (physical page 0). Device memory mapping registers that are used to link logical memory accesses into physical memory accesses are described in Table Section 2.1.1b. After the device memory mapping register has been loaded with the starting physical page number, the device can access physical memory starting at that page and through the next 255 contiguous physical memory pages.

The Extended Memory Management is disabled by the PSRESET signal issued at power-up or under software control. All memory accesses are directed to the lower 255 physical pages. to activate Extended Memory Management, the device memory mapping registers are first loaded with their starting page numbers and then the Enable Extended Memory Management Bit is set in the Extended Memory Status Register.

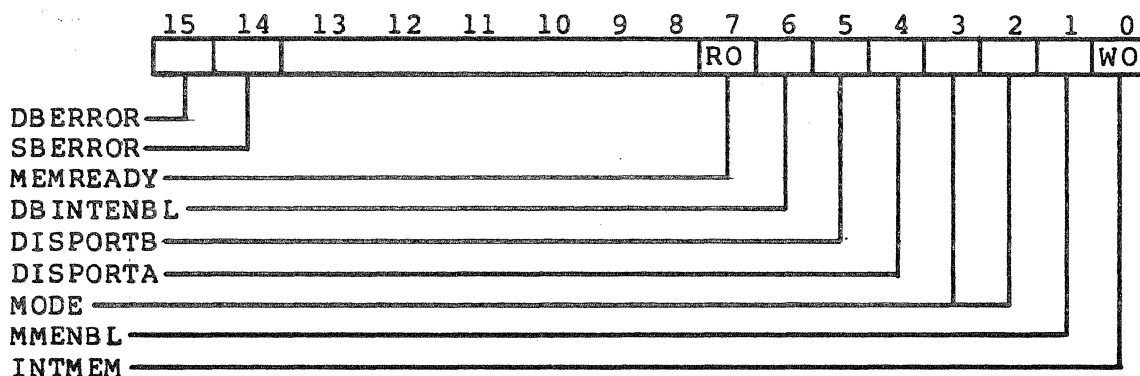
Extended Memory will be initialized when either of the following conditions occur:

1. Power is applied to the Picture System.
2. The Memory Reset Button on the Power Controller is pressed.
3. The Memory Reset Bit in the Extended Memory Status register is set.

The Extended Memory is initialized by the hardware. the hardware will first write zeros into all the memory locations and then read back all locations checking for errors. If any errors are found the respective LED and Error Status Bit will be set.

a. Extended Memory Status Register (EXMSR)-SCB:177410

The EXMSR is used to provide status and control for the Extended Memory subsystem.



BIT	NAME	FUNCTION
15	DBERROR	A double bit error has been detected. Cleared by PSRESET or by writing a one into this bit.
14	SBERROR	A single bit error has been detected and corrected. Cleared by PSRESET or by writing a one into this bit.
7	MEMREADY	Memory has been initialized and is ready.
6	DBINTENBL	Set to one to generate a System Interrupt when a double bit error is detected. Cleared by PSRESET.

- 5 DISPORTB Diagnostic Mode. When this bit is set, Memory Port B is disabled. Cleared by PSRESET.
- 4 DISPORTA Diagnostic Mode. When this bit is set, Memory Port A is disabled. Cleared by PSRESET.
- 3-2 MODE These bits determine the current mode for error correction (detection always on). Cleared to 00 by PSRESET.
- 00 Normal mode error correction on.
- 01 Undefined.
- 10 Error correction off and memory accesses will be to the lower 16 bits.
- 11 Error correction off and memory accesses will be to the upper 6 bits (error correction bits). These bits are right justified.
- 1 MMENBL Set to one to enable Extended Memory Management. Cleared by PSRESET.
- 0 INTEMEM Initialize memory when a one is written into this location.

b. Device Extended Memory Mapping Registers (EXMMR)-SCB:177400-177407

These registers contain the starting physical page numbers used to map logical memory access into physical memory accesses for each Picture System device.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
177407	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177406	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177405	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177404	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177403	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177402	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177401	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
177400	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///	///
											MAP IN MAP OUT REFRESH CONTROLLER DIO DMA					

2.1.2 System Control Block (SCB)

The SCB is a 256-word block of Picture Memory (177400-177777) reserved for device control registers, Matrix Arithmetic Processor registers, Maintenance registers, etc. Figure 2.1-3 shows the SCB and the blocks of registers contained within the SCB. The individual register specifications are described in detail throughout this chapter.

177400 : 177727	Picture System Device Registers
177730 : 177743	Picture Generator Maintenance and Control Registers
177744 177745	Real Time Clock Control Registers
177746	DMA 2 PS Address
177747	DMA 1 PS Address
177750 : 177757	Picture Processor Maintenance and Control Registers
177760 : 177767	System and Device Interrupt Control Registers
177770 : 177777	Passive Device Input and/or Output Ports

Figure 2.1-3
Register Blocks within the SCB

2.1.3 Picture System Data Transfers

All data transfers to or from the Picture System or from device to device within the Picture System are performed by an Active device transferring data to a Passive device or by an Active device transferring data from a Passive device. An Active device is one which can initiate and control the transfer of data to or from a Passive device. A Passive device is one which can accept data from, or provide data for transfer to, an Active device. Associated with each Active/Passive data transfer is a request/acknowledge system. This system allows each word of data requested for transfer by an Active device to be transferred to or from the Passive device before the next data transfer request will be issued by the Active device. Associated with each Active device is a Picture System address register which is used to address the Passive device to be used for the data transfer.

All Picture System devices are either Active or Passive. Although some devices may be either Active or Passive, no device can be both Active and Passive simultaneously. Most Picture System devices are Passive. An example of a Passive device is the Picture Memory which waits "Passively" until requested by an Active device to transfer a word of data onto the PSBUS. The memory will transfer the data onto the bus and will then wait for the next data transfer request. More than one Active device may address a Passive device at any given time. There are five Active devices in the standard Picture System:

1. The PDP-11/Picture System Direct I/O Interface
2. The PDP-11/Picture System DMA I/O Interface
3. The Picture Processor MAP Input Controller
4. The Picture Processor MAP Output Formatter
5. The Picture Generator Refresh Controller (PS2) or the Refresh and Device Processor (MPS)

Each of these Active devices has the ability to initiate and control the transfer of data to or from a Passive device. The Picture System Address Register associated with each Active device is used to address the Passive device which is the partner in the data transfer. In the case of Picture Memory, this Address Register is used to indicate the location data are to be transferred to or from, and the Address Register is incremented by the Active device for the next transfer request. If the Address Register is addressing one of the Passive I/O ports (SCB:177770-177777), then the data transfer occurs at the rate the Passive device can accept the data. Note that the incrementation of the Address Register is inhibited so that the next data transfer is to the same Passive I/O port.

2.2 Picture System/PDP-11 I/O Interface

The PDP-11 computer communicates with the Picture System over two I/O paths:

1. The Direct I/O (DIO) path,
2. The Direct Memory Access (DMA) path.

Using the Direct I/O path, the PDP-11 computer can transfer a single word of data to or from the Picture System. This provides convenient access to the registers of the SCB from the PDP-11.

Using the DMA path, blocks of data may be transferred to or from the Picture System without direct supervision by the PDP-11 CPU.

The Direct I/O and DMA input and output paths to the Picture System are Active devices. Each can initiate and control the transfer of data via the control registers which reside in the PDP-11 and the SCB of the Picture System. Additionally, the DMA input path to the PDP-11 may be used as a Passive device with input being directed by an Active device to the DMA Passive Input Port (SCB:177770).

The Direct I/O and DMA registers are described in the following sections.

2.2.1 Direct I/O Registers

a. Picture System Data Register (PSDATA): 767660

The PSDATA is a 16-bit Read/Write register used to write (or read) data, one word at a time, to the Picture System. Each word written to (or read from) the Picture System is transferred to/from the PSMEM location currently addressed by the DIOPSA register (see below). Each time the PSDATA register is referenced, the DIOPSA is incremented by one. If the reference was a read, the contents of the next sequential location are then transferred to the PSDATA register. If the reference was a write, the contents of the PSDATA register are then transferred to the location specified by the DIOPSA. In either case, after the reference to the PSDATA register, the DIOPSA is incremented. This allows sequential locations of Picture System memory, or the SCB, to be easily read or written. The incrementation of the DIOPSA register may be inhibited by setting the PSAHOLD bit in the IOST register (see below) or by having the DIOPSA register address one of the eight Passive I/O ports in the SCB (177770-177777). Data in this register is valid only upon completion of the Direct I/O transfer as indicated by the DIOREADY bit in the IOST register (see below). The PSDATA is a word register and should not be addressed using byte instructions.

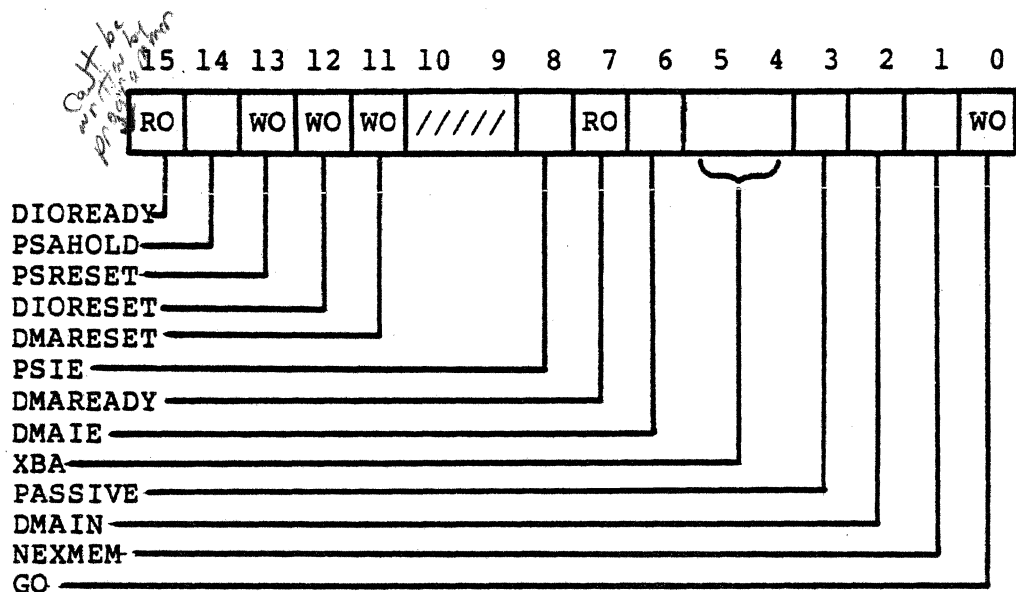
b. DIO Picture System Address Register (DIOPSA): 767662

The DIOPSA is a 16-bit write-only register. It is initially loaded with the Picture System memory address to be read or written by the Direct I/O path. Thereafter, each read or write reference to the PSDATA register will cause the DIOPSA to be incremented by one. The incrementation of this register can be inhibited by setting the PSAHOLD bit in the IOST register (see below) or by addressing one of the eight Passive I/O ports in the SCB (177770-177777). The contents of the DIOPSA can only be read in the following manner:

1. Read the DIOPSA register (the contents read will be undefined). This indicates that the current DIO Picture System address is to be transferred to the PSDATA register. *This can then be read in the PSDATA register*
2. Wait for the Direct I/O transfer to complete by testing the DIOREADY bit in the IOST register.
3. Read the PSDATA register. This register will now contain the location in the Picture System currently being addressed by the Direct I/O path. In this case, reading PSDATA will not cause the DIOPSA to increment.

The DIOPSA is a word register and should not be addressed using byte instructions. It is cleared by PSRESET.

2.2.2 I/O Status Register (IOST): 767670



This register is used to provide status indicators of the Direct I/O and DMA I/O operations, to reset the Picture System and the I/O channel to an initial state and to initiate a DMA transfer to or from the Picture System. This register is not byte addressable and must be addressed using word instructions.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	DIOREADY	Cleared upon initiation of a Direct I/O transfer (i.e., any reference to the PSDATA or DIOPSA registers). Set to indicate that the Direct I/O channel is ready to perform another operation and that the previous operation has completed. Set by INIT, PSRESET or DIORESET. Read only.
14	PSAHOLD	Set to inhibit incrementing of the DIOPSA register so that subsequent data transferred to/from the Picture System are to/from the same Picture System memory location. Cleared by INIT or PSRESET. Read/Write.

PS2/MPS Reference Manual
Hardware Details

13	PSRESET	When set, causes a pulse to the Picture System which resets all Picture System registers and I/O processes to their initial state. This provides a mechanism for initializing the Picture System. Setting PSRESET causes a RESET pulse to be issued to the Picture System (equivalent to the RESET button on the backpanel) and also performs the functions of DIORESET and DMARESET. Write only. Always reads as a zero.
12	DIORESET	When set, causes all Direct I/O processes to and from the Picture System to be reset to their initial state. This is used to initialize the Direct I/O processes. Write only. Always reads as a zero.
11	DMARESET	When set, causes all Direct Memory Access I/O processes to and from the Picture System to be reset to their initial state. This is used to initialize the DMA I/O processes. Write only. Always reads as a zero.
10-9	not used	
8	PSIE	PSIE (Picture System Interrupt Enable) is set to allow any interrupt request from the Picture System to cause an interrupt at the PDP-11. This bit is used in conjunction with the interrupt enable registers of the Picture System (see Section 2.5, Interrupt Control, for details). Cleared by INIT or PSRESET. Read/Write.
7	DMAREADY	Set to indicate that the DMA has completed the last transfer and is ready to accept a new command. Set by INIT, PSRESET, DMARESET or

2-15

end of DMA

		NEXMEM; cleared by GO. Set on word count overflow of the DMAWC register (177777 => 000000). Causes an interrupt if bit 6 (IE) is set. Forces the DMA to release control of the UNIBUS and prevents further DMA cycles. Read only.
6	DMAIE	DMAIE (DMA Interrupt Enable) is set to allow DMAREADY or NEXMEM = 1 to cause an interrupt. Cleared by INIT or PSRESET. Read/Write.
5-4	XBA	XBA (Extended Bus Address) specifies the UNIBUS address bits 17 and 16 that, in conjunction with DMABA, form the 18-bit address to be used for direct memory transfers. Cleared by INIT or PSRESET. XBA17 and XBA16 increment when DMABA overflows. Read/Write.
3	PASSIVE	This bit is used to set the DMA into a mode of operation where it will function as a <u>Passive device for transfer of data from the PICTURE SYSTEM to the PDP-11.</u> All data directed to the DMA Passive Input Port register in the SCB will be transferred to the PDP-11 and stored at the current address specified by DMABA and XBA<17:16>. The DMA can function as a Passive device if data are to be transferred back to the PDP-11. Before the passive input transfer can be performed, the number of words to be transferred (word count) and the base address in PDP-11 memory must be set in the DMAWC and DMABA registers and the transfer initiated by setting the GO bit. Cleared by INIT or PSRESET. Read/Write.

2	DMAIN	DMAIN (DMA Input) is used to indicate the current transfer direction of the DMA. Cleared by INIT or PSRESET. Read/Write. If DMAIN=0, the data transfer direction is out--from the PDP-11 to the Picture System. If DMAIN=1, the data transfer is in--from the Picture System to the the PDP-11.
1	NEXMEM	NEXMEM (Nonexistent Memory) is set to indicate that a UNIBUS master, the DMA, did not receive a SSYN response 10 usec after asserting MSYN. Cleared by INIT, PSRESET, DMARESET or loading with a 0; cannot be loaded with a 1.
0	GO	This bit is set to initiate the DMA transfer. Setting this bit clears DMAREADY. Note that no other bits in this word are able to be modified by the same instruction which causes this bit to be set. Write only. Always reads as a 0.

2.2.3 Direct Memory Access Registers

This section deals with the mechanism used to pass blocks of data between the PDP-11 and the Picture System. Note that blocks of data can be transferred in either direction.

The block data transfer path is a Direct Memory Access interface unit. To pass data to or from the Picture System, a block of PDP-11 memory containing the data, or which will receive the data, is specified by loading registers in the DMA interface. The location in the Picture System to or from which words are to be transferred must also be specified by loading the DMAPSA (see below).

a. DMA Word Count Register (DMAWC): 767664

The DMAWC is a 16-bit register. It is initially loaded with the two's complement of the number of words to be transferred and increments up towards zero after each bus cycle. When overflow occurs (all 1's to all 0's), the DMA READY bit of the IOST is set and the bus cycles stop. DMAWC is a word register; byte instructions should not be used when loading this register. Cleared by INIT or PSRESET.

b. DMA Bus Address Register (DMABA): 767666

The DMABA is a 15-bit R/W register. Bit 0 is always a zero, and is a read-only bit. Along with bits 5 and 4 of the IOST (XBA17 and XBA16), the DMABA is used to specify the address used when the DMA accesses the UNIBUS. The register is incremented (by 2) after each bus access, advancing the address to the next sequential word location on the bus. If DMABA overflows (177776 to 0), the bus extension bits of the IOST are incremented and the transfer continues. DMABA is a word register; byte instructions should not be used when loading this register. Cleared by INIT or PSRESET.

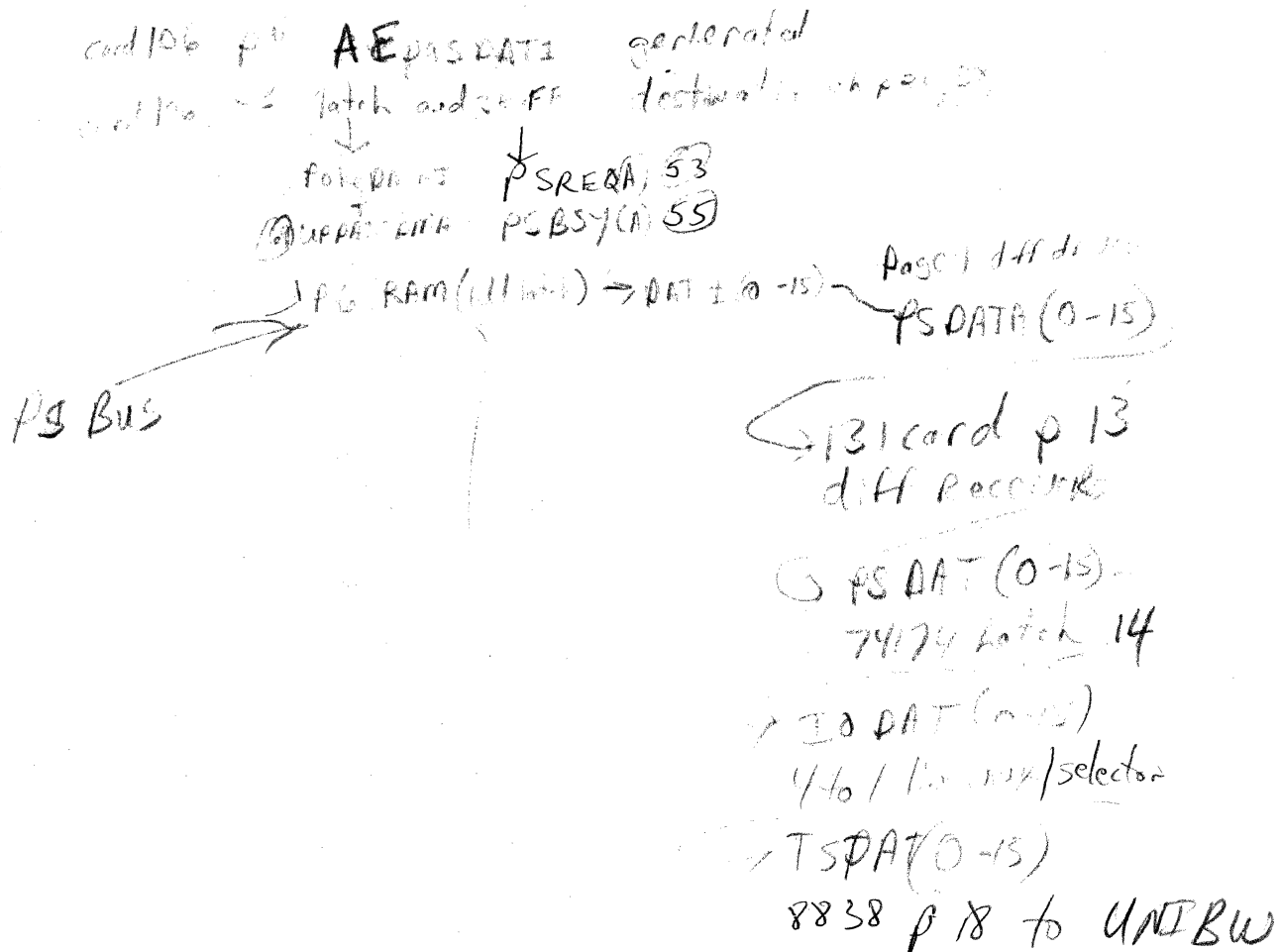
c. DMA Picture System Address Register
(DMAPSA)-SCB:177747

The DMA Picture System Address Register is a 16-bit R/W register in the Picture System SCB, which indicates the next word of Picture System memory to be accessed, whether as a read or write or an input to, or output from, the memory. After each word is transferred, this address is then incremented so that the next word transferred will be to (or from) the next sequential

location. If the DMAPSA is currently addressing one of the eight Passive I/O ports in the SCB (177770-177777), incrementation of the DMAPSA will be inhibited. Cleared by PSRESET.

d. DMA Passive Input Port (DMAPIP)-SCB:177770

The DMA Passive Input Port is a 16-bit write-only location in the PICTURE SYSTEM SCB to which data may be directed by an Active device for DMA transfer to the PDP-11. In order to use the DMAPIP correctly, the DMA must be set up and initiated in the PDP-11 and the Active device's address register should then be set to address the DMAPIP using the Direct I/O path to the Picture System. Once this has been done, all subsequent data output by the Active device will be transferred to the PDP-11 without direct intervention by the CPU until the entire block of data which was set up to be transferred has been completed.



2.3 The Picture Processor

The Picture Processor is a microprogrammed, high-speed, digital arithmetic processor capable of accepting two-, three-, or four-dimensional data, transforming the data by a previously loaded or concatenated Transformation Matrix, clipping the transformed data at the boundaries of a six-sided window, performing a perspective division and viewport mapping on the clipped data, and outputting the transformed, clipped and viewport mapped data to the refresh buffer as a portion of a segmented display file for subsequent display on a CRT. There are three basic units of the Picture Processor:

1. The MAP Input Controller
2. The Matrix Arithmetic Processor (MAP)
3. The MAP Output Formatter

These units function together in a synchronous pipelined manner to provide maximum throughput for the Picture Processor.

The Picture Processor is microprogrammed to perform the functions described throughout this section. The Picture Processor is, however, capable of being provided as a general-purpose, microprogrammable digital processor. When used in this manner, the microprogram is loaded, using the Picture Processor Maintenance Registers described in Section 2.3.4.

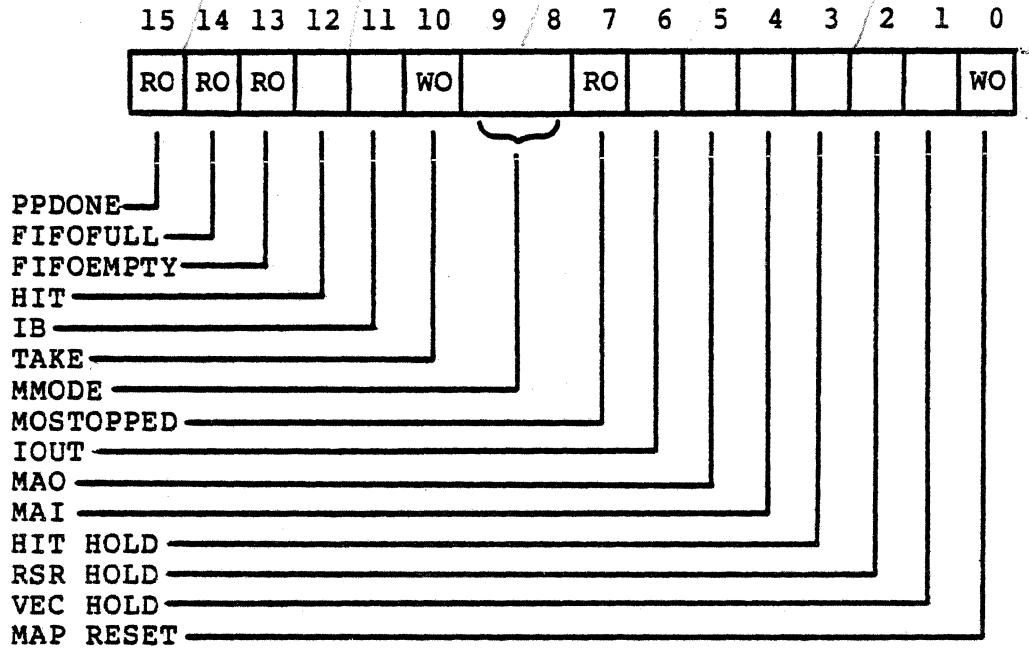
The Picture Processor input, output and status is controlled by eight registers in the SCB. These registers, shown in Figure 2.3-1, are described in the following sections which detail the three basic units of the Picture Processor.

PS2/MPS Reference Manual
Hardware Details

SCB:177750	MAOL
177751	MAOA
177752	MAIA
177753	MSR
177754	MAP
177755	Maintenance
177756	Control
177757	Registers

Figure 2.3-1
Picture Processor I/O Control,
Status and Maintenance Registers

MAP Status Register (MSR)-SCB:177753



The MSR is the basic control register of the Picture Processor. It is used to provide operating mode information to the MAP, and status of the MAP back to the control program.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	PPDONE	PPDONE (Picture Processor Done) is set to indicate that the MAP Output Formatter is finished with its processing and the MAP itself is in the HOLD state waiting for another processing request. PPDONE is also set whenever the MAP is waiting for more data to be input to the MAP Input FIFO regardless of whether the last RSR Command has completed processing. Read only. Set by RESET.
14	FIFOFULL	When this bit is set, it indicates that the MAP input FIFO is full and has 4 words of data buffered for input to the

		MAP. Read only. Cleared by RESET.
13	FIFOEMPTY	When this bit is set, it indicates that the MAP input FIFO is empty. Read only. Set by RESET. The MAP input FIFO is emptied (and this bit is set) by loading the MAP Active Input Address register.
12	HIT	This bit is cleared at the beginning of the processing of each input vector. While processing, if any part of the vector is within the window, HIT is set. If the HIT Interrupt Enable (see Section 2.5) is set, this bit will cause an interrupt to occur. Read only unless the write is enabled by the TAKE bit (10) being set. Cleared by RESET.
11	IB <i>Input base</i>	IB is used to designate where the INPUT, SAVE, BASE and MOUT x,y,z and w coordinates are located within the MAP Register File. Each time a new vector is input, IB is complemented. INPUT, SAVE, BASE and MOUT X,Y,Z and W are valid only in between and after successive vector commands (i.e., a Matrix concatenation destroys the current INPUT, BASE, SAVE and MOUT X,Y,Z and W coordinates. IB is cleared by RESET and can be set or cleared from software. If IB=0 before INPUT then: ABSOLUTE - Data input written into MAP 0-3. RELATIVE - Previous information in MAP 4-7. New input plus previous written into MAP 0-3.

ORIGIN OFFSET -
BASE information in MAP 0-3.
New input plus BASE written
into MAP 0-3. BASE informa-
tion moved to MAP 4-7.
SET BASE -
Data input written into MAP
0-3 and into MAP 4-7.

If IB=1 before INPUT then:

ABSOLUTE -
Data input written into MAP
4-7.
RELATIVE -
Previous information in MAP
0-3. New input plus previous
written into MAP 4-7.
ORIGIN OFFSET -
BASE information in MAP 4-7.
New input plus BASE written
into MAP 4-7. BASE informa-
tion moved to MAP 0-3.
SET BASE -
Data input written into MAP
0-3 and into MAP 4-7.

Read only unless the write is
enabled by the TAKE bit (10)
being set. Cleared by RESET.

10

TAKE

Handwritten:
this bit
write hit
or IB
100

This bit is used to enable the
writing of bits 11 and 12 (IB
and HIT). If this bit is set
when the MSR is written, then
bits 11 and 12 are written
into the register; otherwise
bits 11 and 12 of the MSR re-
main unmodified. Write-only.
Always reads as a zero.

9,8

MMODE

The MMODE (MAP Mode) bits are
used to control the mode in
which the MAP processes data
and the Output Formatter out-
puts the data. Set to 00 by
RESET. The bits indicate the
mode as follows:

00

Display Mode
In this mode, all two-,
three-, or four-dimensional

data are transformed, normalized, clipped, viewport mapped and output formatted for input to the Picture Generator. All Pass Formatted data will be formatted for input to the Picture Generator.

01

Sixteen-Bit Precision Mode

In this mode, all two-, three-, or four-dimensional data are transformed, normalized, clipped, viewport mapped and formatted as a linear display list with an RSR (see Section 2.3.2.2) for each set of X,Y and Z coordinates output. All data that are to be merely passed through the MAP are output unmodified. All Control Commands are executed by the MAP and the data output according to the command. All data are output as absolute 3D data (i.e., X,Y,Z). All "Pass" data are output in 3D Pass with only the first and second words being valid (i.e., the words corresponding to the X and Y coordinates).

10

Transform/Normalize Mode

In this mode, all two-, three-, or four-dimensional data are transformed and normalized only and output as X,Y,Z and W coordinates--each significant to 16-bits. The clipping, perspective division and viewport mapping is not performed upon data when in this mode. All other forms of data (Pass, etc.) should not be used when in this mode.

11

Transform Only Mode

In this mode, all two-, three-, or four-dimensional data are transformed only and output as X,Y,Z and W coordinates--each significant to 16-bits. The clipping, perspective division and viewport

mapping is not performed upon data when in this mode. All other forms of data (Pass, etc.) should not be used when in this mode. It should be noted while in this mode it is possible to have the matrix multiplication produce overflow such that the data output by the MAP as 16-bits does not represent the same number kept internally in the MAP to 24 bits of precision. This overflow can be read and compensated for by the control program in the host computer.

7

MOSTOPPED

When MAP Output Stopped (MOSTOPPED) is set, it indicates that output of the MAP has stopped since the MAP Active Output Address register is equal to the MAP Active Output Limit register. This condition indicates that the area of Picture System memory allocated for output data (i.e., refresh buffer, etc.) is full. The condition can only be dismissed by resetting either the MAP Active Output Address (MAOA) or the MAP Active Output Limit (MAOL) register. The setting of this bit will cause an interrupt to the host computer if the MOSTOP interrupt enable bit is set in the System Interrupt Enable register (see Section 2.5). Read only. Cleared by resetting MAOA or MAOL so that MAOA is not equal to MAOL. Set by MAOA=MAOL and RESET.

6

IOUT

When Inhibit Output (IOUT) is set, it inhibits the MAP Output Formatter from outputting any further data. Input to the MAP may proceed as if output were not inhibited. This bit is useful in "hit testing" where data passing through a hit window would appear mis-

placed if output by the Picture Processor for display. Cleared by RESET.

5 MAO

When the MAP Active Output (MAO) bit is set, the MAP Output Formatter functions as an Active device, initiating and controlling the output of data from the MAP to the Passive device currently selected by the MAP Active Output Address register--typically, Picture System memory. Cleared by RESET.

4 MAI

When the MAP Active Input (MAI) bit is set, the MAP Input Controller functions as an Active device filling the MAP input FIFO with data from the Passive device currently selected by the MAP Active Input Address register. Clearing this bit is the only way in which to halt the MAP Input Controller. Cleared by RESET.

3 HIT HOLD

When this bit is set, the MAP will stop the processing of new data after a hit occurs, as indicated by the HIT bit (12). Processing of new data by the MAP can be continued only by acknowledging the HIT Request bit in the System Request register (see Section 2.5). Cleared by RESET.

2 RSR HOLD

When this bit is set, the MAP will stop taking data from the input FIFO after execution of the current MAP RSR command has completed. The next RSR command to be processed will be the first item in the FIFO, if any data resides in the FIFO. After this bit is set, PPDONE (bit 15) indicates that the Picture Processor has completed the last RSR operation and is now in the HOLD state.

Cleared by RESET.

1 VEC HOLD

When this bit is set, the MAP will stop taking data from the input FIFO when processing for the current vector is complete. After this bit is set, PPDONE (bit 15) indicates that the Picture Processor has completed the last vector and is now in the HOLD state. Cleared by RESET.

0 MAP RESET

When this bit is set, it causes a RESET signal to be issued to the MAP, resetting it to its initial power-up state. Write only. Always reads as a zero.

2.3.1 The MAP Input Controller

The MAP Input Controller is a Picture System device which can function as either an Active or Passive device. When functioning as either, the operation of the Input Controller is comparatively simple--it transfers 16-bit words of data, either actively or passively received, to a four-word FIFO (First In First Out) buffer. As the FIFO becomes full, the Input Controller waits for the MAP to empty a word of data from the FIFO before the next word of data is transferred to the FIFO. There are two status indicators of the current state of the MAP input FIFO. These indicators are "FIFO FULL" and "FIFO EMPTY" and exist as bits in the MAP Status Register.

As previously stated, the MAP Input Controller can function as either an Active or Passive device. The Active/Passive state of the Input Controller is selected by a bit in the MAP Status Register (MSR). When the MAP Input Controller is selected as an Active device, there is an associated MAP Active Input Address register (MAIA) which holds the address of the location in PSMEM from which the next word of data to be sent to the MAP Input FIFO is taken.

a. MAP Active Input Address (MAIA)-SCB:177752

The MAIA is a 16-bit R/W register which holds the PSMEM address to be used to fetch the next word of data for input to the MAP. After each word of data is transferred to the MAP Input FIFO, this address is incremented by the MAP Input Controller so that the next word of data is taken from the next sequential address of Picture System memory. This register should not be used to address any Passive device other than Picture System memory. Incrementation of this device cannot be inhibited by addressing one of the eight Passive I/O Ports (SCB:177770-177777).

If the MAP Input Controller is not selected as an Active device, it is, therefore, a Passive device and all data directed to its Passive Input Port (MPIP, SCB:177777) is stored in the MAP input FIFO for input to the MAP.

b. MAP Passive Input Port (MPIP)-SCB:177777

The MAP Passive Input Port is a 16-bit write-only location in the Picture System SCB to which data may be directed by an Active device for input to the MAP. In order to use the MPIP, the Active device's output address register should be set to address the MPIP using the Direct I/O path to the Picture System and the MAP Input Controller selected for Passive input. Once this has been done, all subsequent data output by the Active device will be transferred to the MAP without direct intervention by the host CPU.

2.3.2 The Matrix Arithmetic Processor (MAP)

The Matrix Arithmetic Processor consists of a Transformation Matrix, Transformation Matrix Stack, Parameter Register File, and Arithmetic Unit.

The Transformation Matrix is a 4x4 element matrix, where each element is a 16-bit word. This 4x4 matrix is used to transform object coordinate data. It can also be concatenated with other 4x4 matrices to obtain a combined transformation.

The Transformation Matrix Stack is a storage area where up to fourteen 4x4 element matrices may be "stacked" or saved for future recall.

The Parameter Register File is an array of registers into which parameters specifying viewport boundaries, scale factors, etc. are stored and may be retrieved.

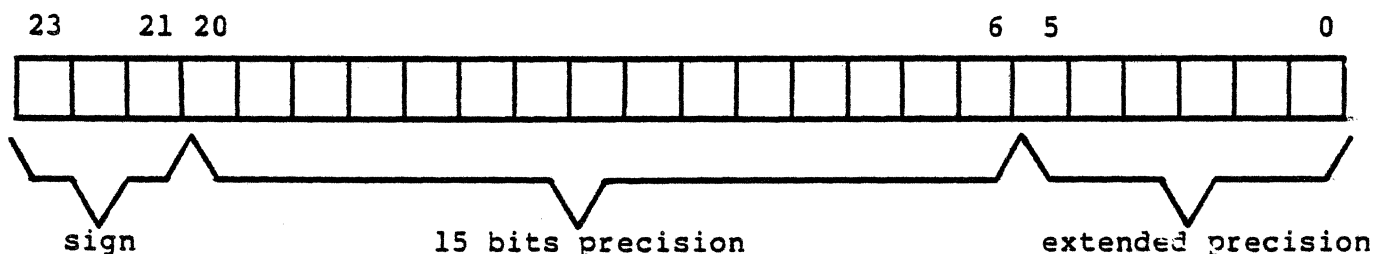
The Arithmetic Unit performs all scalar, vector and matrix arithmetic operations in the Picture Processor. This includes subtraction, addition, multiplication, division and normalization.

The MAP utilizes these units to perform digital operations on the data received from the host computer.

2.3.2.1 MAP Internal Registers

This section describes the 256 registers that are internal to the MAP and are used to contain the Transformation Matrix, Matrix Stack, parameters for various functions, and working storage during command execution. Figure 2.3-2 shows these registers and the addresses assigned to them. These addresses are specified using the "Operand" field of the RSR Control Commands (see Section 2.3.2.2).

Each of the 256 MAP registers are 24-bit registers of the following format:



The high three bits (23-21) represent the sign of the number. The lower 21 bits (20-0) are used to represent the number to 21 bits of significance. When data are input to or output from the MAP, the words transferred as 16 bits (1 bit sign, 15 bits data) to or from bits 21-6 of the specified MAP register. When the transfer is from a MAP register, bits 21-6 are transferred. When the transfer is to a MAP register, bits 5-0 are loaded with 0, bits 21-6 are loaded with the 16-bits of data transferred to the MAP, bits 23-22 are sign extended from bit 21. All 24 bits of each of the 256 MAP registers may be LOADED or STORED by using the extended LOAD and STORE commands (see Section 2.3.2.2).

It should be noted that all MAP arithmetic operations are performed treating data as two's complement signed (fixed point) fractions. Since the basic word length of the Picture System is 16 bits, the algebraically largest number that can be represented is $1-2^{15}$ (i.e., 32767 decimal or 77777 octal), and the algebraically smallest number that can be represented is -1 (-32768 decimal or 100000 octal). In binary notation (with the binary point separating the sign bit from the fraction):

0.111111... is the algebraically largest number.
0.000000... is the unique representation for zero.
1.000000... is the algebraically smallest number.

PS2/MPS Reference Manual
Hardware Details

The closest approximation to +1 is the fraction 0.111111..., which is close enough to +1 for practical cases. Thus, the MAP internal representation of 24 bits is an extension of the 16-bit two's complement fractions described above. Three bits of sign (23-21) are used since during matrix multiplication four numbers are summed together, which can require up to 3 bits to maintain the correct sign of the summation. Normalization is always performed to the binary point.

0	INPUTx/BASEx
1	INPUTy/BASEy
2	INPUTz/BASEz
3	INPUTw/BASEw
4	BASEx/INPUTx
5	BASEy/INPUTy
6	BASEz/INPUTz
7	BASEw/INPUTw
10	MCUTx/SAVEx
11	MOUTy/SAVEy
12	MOUTz/SAVEz
13	MOUTw/SAVEw
14	SAVEx/MOUTx
15	SAVEy/MOUTy
16	SAVEz/MOUTz
17	SAVEw/MOUTw
20	Viewport X 1/2 Size
21	Viewport X Center
22	Viewport Y 1/2 Size
23	Viewport Y Center
24	Viewport Z Size
25	Viewport Z Front
26	Working Register
27	Transformation Matrix Address
30	NEWCLIPx/CLIPSAVEx
31	NEWCLIPy/CLIPSAVEy
32	NEWCLIPz/CLIPSAVEz
33	NEWCLIPw/CLIPSAVEw
34	CLIPSAVEx/NEWCLIPx
35	CLIPSAVEy/NEWCLIPy
36	CLIPSAVEz/NEWCLIPz
37	CLIPSAVEw/NEWCLIPw
40	Transformation Matrix
:	and Matrix Stack for
377	14 4x4 Matrices

Figure 2.3-2
MAP Parameter Register File

In Figure 2.3-2, the "/" designation (i.e., INPUT/BASE) is used to indicate that the particular MAP register may be used for either the INPUT or BASE register depending upon the state of the IB bit in the MAP Status Register. In all of the "toggled" registers (i.e., INPUTx/BASEx) the first register name (INPUTx) is selected when IB=1 and the second name (BASEx) is selected when IB=0. The state of IB is toggled immediately after each new vector is input to the MAP. Following is a description of each register and its usage.

INPUT (INPUT), Registers 0-3 or 4-7

The INPUTx, INPUTy, INPUTz and INPUTw registers are used to hold the X,Y,Z and W data input to the MAP for 2D, 3D or 4D DRAW commands. Alternates with BASE.

BASE (BASE), Registers 4-7 or 0-3

The BASEx, BASEy, BASEz and BASEw registers are used to hold the current X,Y,Z and W base coordinates for 2D, 3D or 4D DRAW commands. For Relative and Origin Offset draws, the BASE register is used to compute the absolute coordinates for the input data. Alternates with INPUT.

Matrix Output (MOUT), Registers 10-13 or 14-17

The MOUTx, MOUTy, MOUTz and MOUTw registers are used to hold the normalized matrix output X,Y,Z and W coordinates at the completion of a 2DDRAW, 3DDRAW or 4DDRAW command execution. MOUT contains the data as it exists after it has been multiplied by the Transformation Matrix, but prior to any clipping that may have taken place. Alternates with SAVE.

SAVE (SAVE), Registers 14-17 or 10-13

The SAVEx, SAVEy, SAVEz and SAVEw registers are used to hold the unclipped X,Y,Z and W coordinates of the previous point. The unclipped coordinates are necessary to properly compute the next intersection of the clipping plane if required. Alternates with MOUT.

Viewport X Half-Size (VIEWXH), Register 20

This register is used to specify the half-width of the X viewport boundaries to which data that lies within the clipping boundaries will be mapped by the viewport mapping process. This value is typically 0-2K, causing the data to be mapped to the boundaries of the Picture Display. If a negative value is placed here, the mapping will be such that a mirror image is produced about the y-axis of the viewport.

Viewport X Center (VIEWXC), Register 21

This register is used to specify the center of the viewport in the x-axis. This value, in conjunction

with VIEWXH, is used to specify the left and right viewport boundaries. The boundaries are computed as follows:

Viewport Left boundary = VIEWXC - VIEWXH
Viewport Right boundary = VIEWXC + VIEWXH

Viewport Y Half-Size (VIEWYH), Register 22

This register is used to specify the half-width of the Y viewport boundaries to which data that lies within the clipping boundaries will be mapped by the viewport mapping process. This value is typically 0-2K, causing the data to be mapped to the boundaries of the Picture Display. If a negative value is placed here, the mapping will be such that a mirror image is produced about the x-axis of the viewport.

Viewport Y Center (VIEWYC), Register 23

This register is used to specify the center of the viewport in the y-axis. This value, in conjunction with VIEWYH, is used to specify the bottom and top viewport boundaries. The boundaries are computed as follows:

Viewport Bottom boundary = VIEWYC - VIEWYH
Viewport Top boundary = VIEWYC + VIEWYH

Viewport Z Size (VIEWZS), Register 24

This register is used to specify the size of the Z viewport boundaries to which data that lies within the clipping boundaries will be mapped by the viewport mapping process. This value is typically 63-0, causing the data to be mapped so that it may be displayed as depth-cued information.

Viewport Z Front (VIEWZF), Register 25

This register is used to specify the front of the viewport in the Z-axis. This value, in conjunction with VIEWZS, is used to specify the hither and yon viewport intensities. The intensities are computed as follows:

Viewport Hither intensity = VIEWZF
Viewport Yon intensity = VIEWZF + VIEWZS

Working Register, Register 26

This register is used as a working register whose contents are undefined at any given time.

Transformation Matrix Address (TMADR), Register 27

This register is used to contain the current MAP register address (0-377) which is to be used as the Transformation Matrix. This register is used as a Matrix Stack pointer in that a Matrix PUSH (or Matrix POP) causes the address to increment (or decrement) by 16. This register always contains the address of the 16th element of the current Transformation Matrix.

New Clip (NC), Registers 30-33 or 34-37

The NCx, NCy, NCz and NCw registers are used to contain the clipped X,Y,Z and W coordinates of the new clipped point if the point required clipping as indicated by the SV and NV bits of the MAP Maintenance Status Register (see Section 2.3.4). Alternates with CSAVE.

Clip Save (CSAVE), Registers 34-37 or 30-33

The CSAVE_x, CSAVE_y, CSAVE_z, CSAVE_w registers are used to contain the clipped X,Y,Z and W coordinates of the previous point. Alternates with NC.

Transformation Matrix (TM), Registers 40-377

The Transformation Matrix is used to contain the 4x4 matrix by which all 2D, 3D and 4D coordinates are transformed. The Transformation Matrix may be any 16 consecutive MAP registers from 40-377. The Transformation Matrix must, however, begin on an even 16-word boundary, thus restricting its beginning address to MAP register 40, 60, 100, 120...360. The address of the current Transformation Matrix is always indicated by the Transformation Matrix Address (TMADR), register 27. Matrices are stacked in the Picture System by copying the current contents of the TM to TM+16 and then incrementing TMADR by 16. See Section 2.3.2.2 for details of the commands which modify the TM. Although allocated as Matrix Stack space, MAP registers 40-377 need not be used only for matrix storage, but may also be used to store other data or as auxiliary (not Matrix) stack area (see Section 2.3.2.2).

2.3.2.2 Picture Processor Commands

In order to properly process data sent to the Picture Processor, a command must be sent to the MAP indicating the operation to be performed. The Repeat Status Register (MMRSR-SCB:177755) is used to supply commands to the Picture Processor. It is called a "Repeat" Status Register since portions of its contents may be automatically modified in a predetermined manner after the specified command has been executed. In certain cases, the command is repeated after the modification without required program intervention. The basic format of the RSR command is shown in Figure 2.3-3.

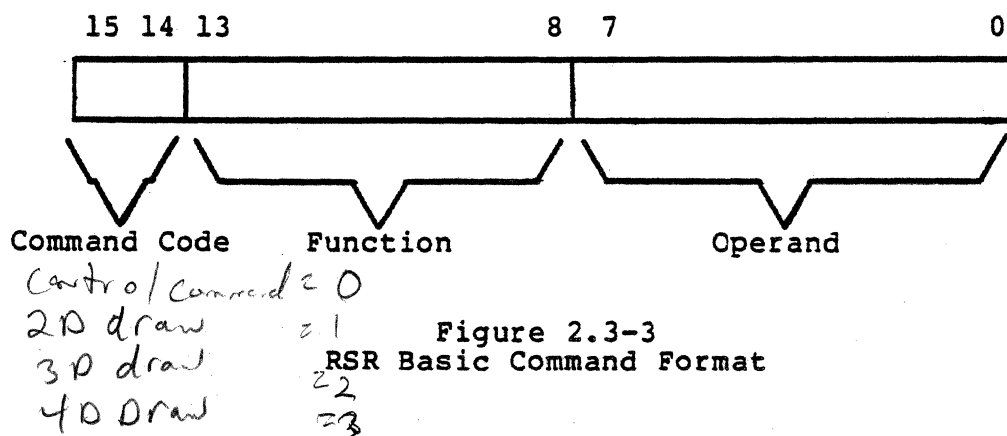


Figure 2.3-3

RSR Basic Command Format

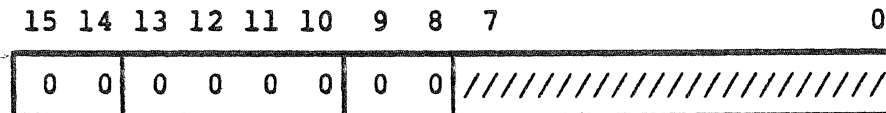
As the figure shows, the basic command format consists of 2 bits of command code (15,14), 6 bits of a Function Field, and 8 bits of an Operand Field. The meaning of the function and operand fields are dependent upon the command currently being processed. There are two basic command codes:

- a. RSR Control Commands
- b. RSR Drawing Commands

These Command Codes and the associated Function and Operand fields are described in detail in the following sections.

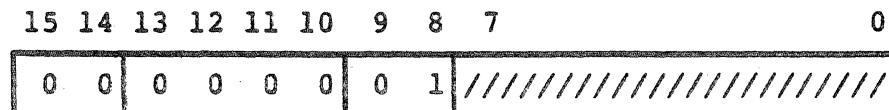
a. RSR Control Commands

HALT



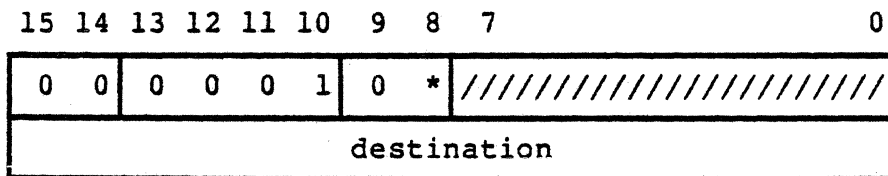
HALT causes the MAP input operation to cease, setting PPDONE (bit 15) in the MAP Status Register (MSR) when the MAP Output Formatter has completed its last operation.

NO-OP



NO-OP causes the next RSR command to be input to the MAP without other action being taken.

JUMP



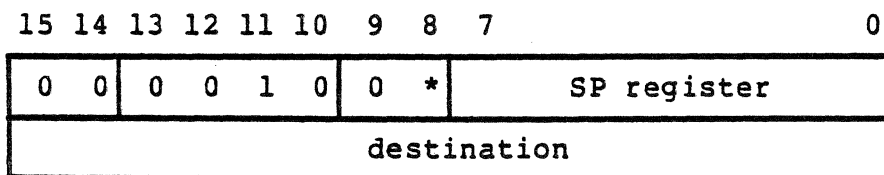
JUMP causes the next word of data input to the MAP to be placed in the MAP Active Input Address (MAIA) register, thereby causing all subsequent input to come from the effective Picture System memory address specified by the destination word. The MAP must be in operation as an Active device for any further data to be transferred from PSMEM. After the Picture System memory address is transferred to the MAIA, the input FIFO is cleared so that the next data processed by the MAP is from the PSMEM location specified. Note that the Picture System memory address specified should contain the next RSR command to be processed. If this command is executed by the MAP when the MAP Input Controller is not selected as an Active device, the JUMP REQ bit is set in the System Interrupt Request register (SCB:177762). This will cause an interrupt to the host computer if the interrupt is currently enabled in the System Interrupt Enable register (SCB:177763). The MAP will then be in the HOLD state with PPDONE (bit 15) in the MSR set.

Bit 8 of the RSR JUMP command is used to determine whether the JUMP is to an absolute address in Picture System memory (Absolute) or to a location relative to the current MAP Active Input Address (Relative).

If bit 8=0: The JUMP is performed as an absolute JUMP to the address specified by destination.

If bit 8=1: The JUMP is performed as a relative JUMP to the effective address computed as MAIA + destination.

PUSHJ



PUSHJ functions exactly as the RSR JUMP command except that the current address in the MAIA is saved in the stack whose stack pointer is specified in the operand field (SP register) of the PUSHJ command. The MAP register being addressed by the operand as the stack pointer is incremented before the current MAIA is saved. The MAP must be in operation as an Active device for any further data to be transferred from PSMEM. After the Picture System memory address is transferred to the MAIA, the input FIFO is cleared so that the next data processed by the MAP is from the PSMEM location specified. Note that the Picture System memory address specified should contain the next RSR command to be processed. If this command is executed by the MAP when the MAP Input Controller is not selected as an Active device, the JUMP REQ bit is set in the System Interrupt Request register (SCB:177762). This will cause an interrupt to the host computer if the interrupt is currently enabled in the System Interrupt Enable register (SCB:177763). The MAP will then be in the HOLD state with PPDONE (bit 15) in the MSR set.

Bit 8 of the RSR PUSHJ command is used to determine whether the JUMP, performed after the current address is pushed onto the stack, is to an absolute address in Picture System memory (Absolute) or to a location relative to the current MAP Active Input Address (Relative).

If bit 8=0: the PUSHJ is performed as an absolute PUSHJ to the address specified by destination.

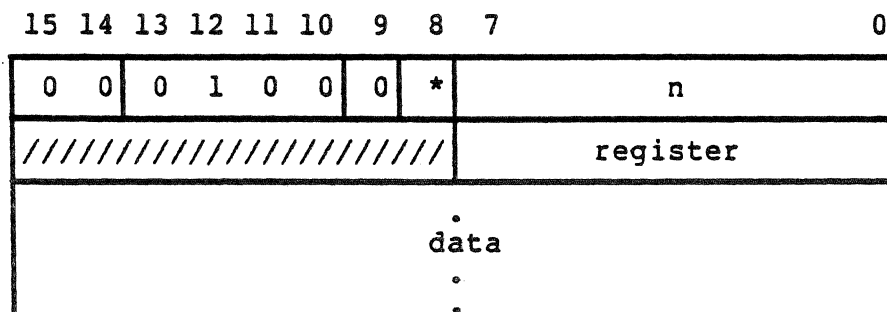
If bit 8=1: The PUSHJ is performed as a relative PUSHJ to the effective address computed as MAIA + destination.

POPJ

15	14	13	12	11	10	9	8	7	0
0	0	0	0	1	1	0	0	SP register	

POPJ causes the current contents of the top of the stack, whose stack pointer is specified in the operand field (SP register), to be placed in the MAIA register. This causes all subsequent input to come from the Picture System memory address that was on the top of the stack. The MAP must be in operation as an Active device for any further data to be transferred from PSMEM. After the Picture System memory address is transferred to the MAIA, the input FIFO is cleared so that the next data processed by the MAP is from the PSMEM location taken from the top of the stack. The MAP register being addressed as the stack pointer is decremented after the MAIA is loaded with the top of the stack. If this command is executed by the MAP when the MAP Input Controller is not selected as an Active device, the JUMP REQ bit is set in the System Interrupt Request register (SCB:177762). This will cause an interrupt to the host computer if the interrupt is currently enabled in the System Interrupt Enable register (SCB:177763). The MAP will then be in the HOLD state with PPDONE (bit 15) in the MSR set.

LOAD



The LOAD command is used to load the 256 MAP registers with data. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be loaded. The next word of data specifies the first MAP register to be loaded. The data which follows is then sequentially loaded into the MAP registers.

If bit 8=0: Each word of data is loaded into each 24-bit MAP register as bits 21-6 of the register. Bits 23-22 are sign extended from the sign of the 16-bit word (bit 21) which was loaded. Bits 5-0 are zeroed.

If bit 8=1: Each 24-bit MAP register is loaded with 24-bits of data taken from the data input in the following manner:

MAP register bit 23 <- data word i bit 1.
MAP register bit 22 <- data word i bit 0.
MAP register <21-6> <- data word i+1 <15-0>.
MAP register <5-0> <- data word i <15-10>.

In this way, 32 bits of data are transferred for each 24-bit MAP register. Bits 9-2 of data word i are discarded.

STORE

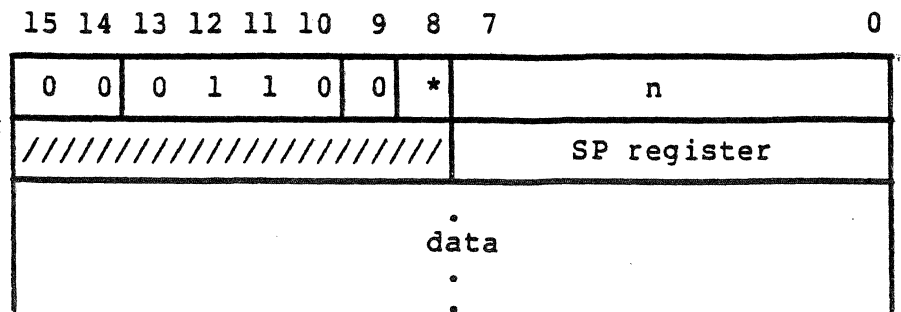
15	14	13	12	11	10	9	8	7		0
0	0	0	1	0	1	0	*		n	
////////////////////									register	

The STORE command is used to store the contents of the 256 MAP registers. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be stored. The next word of data specifies the first MAP register to be stored. The contents of the MAP registers are sequentially transferred to the location specified by the MAP Active Output Address (MAOA) register. Bit 8 of the RSR specifies how the data are to be taken from the MAP registers:

If bit 8=0: Bits 21-6 of each MAP register selected are transferred to the location specified by the MAOA. Note that in this mode, the data stored may not truly reflect the contents of the register. This is especially true if the register contains unnormalized matrix data.

If bit 8=1: Bits 23-22 and 5-0 of each MAP register selected are transferred to bits 1-0 and 15-10 of the location specified by the MAOA. Bits 21-6 of each MAP register are transferred to bits 15-0 of the location specified by MAOA+1.

LOADSTK



The LOADSTK (Load Stack) command is used to load data into a stack area of the 256 MAP registers. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be loaded. The next word of data (SP register) specifies the MAP register which contains the MAP address of the current top of the stack. The data which follows are then loaded into the top n words of the stack within the MAP. The stack pointer is unmodified by this operation.

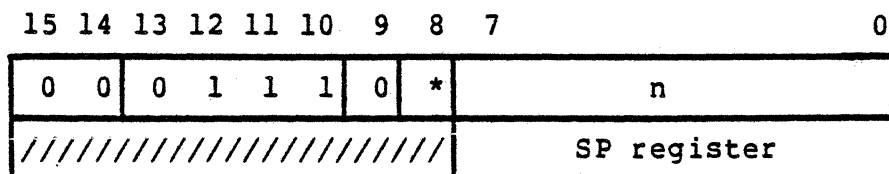
If bit 8=0: Each word of data is loaded into each 24-bit MAP register as bits 21-6 of the register. Bits 23-22 are sign extended from the sign of the 16-bit word (bit 21) which was loaded. Bits 5-0 are zeroed.

If bit 8=1: Each 24-bit MAP register is loaded with 24-bits of data taken from the data input in the following manner:

MAP register bit 23 <- data word i bit 1.
MAP register bit 22 <- data word i bit 0.
MAP register <21-6> <- data word i+1 <15-0>.
MAP register <5-0> <- data word i <15-10>.

In this way, 32 bits of data are transferred for each 24-bit MAP register. Bits 9-2 of data word i are discarded.

STORESTK



The STORESTK (Store Stack) command is used to store data from a stack area of the 256 MAP registers. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be stored. The next word of data (SP register) specifies the MAP register which contains the MAP address of the current top of the stack. The n MAP registers on top of the stack are then output by the MAP. The stack pointer is unmodified by this operation.

If bit 8=0: Bits 21-6 of each MAP register selected are transferred to the location specified by the MAOA. Note that in this mode, the data stored may not truly reflect the contents of the register. This is especially true if the register contains unnormalized matrix data.

If bit 8=1: Bits 23-22 and 5-0 of each MAP register selected are transferred to bits 1-0 and 15-10 of the location specified by the MAOA. Bits 21-6 of each MAP register are transferred to bits 15-0 of the location specified by MAOA+1.

XFER

15	14	13	12	11	10	9	8	7	0
0	0	1	0	0	0	0	*	n	
////////////////////////////////								register-SRC	
////////////////////////////////								register-DES	

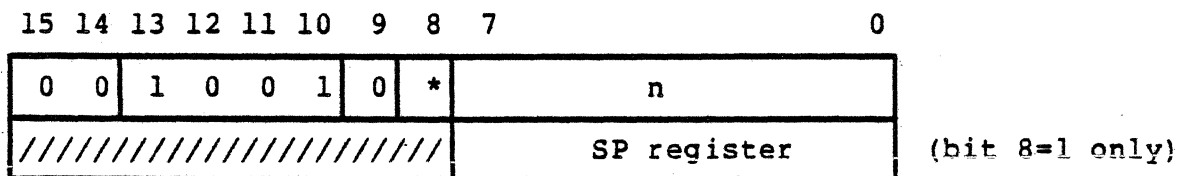
Register file (256 by 24 memory)

The XFER (Transfer) command is used to transfer the contents of the MAP registers from one location to another. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be transferred. The next word of data (register-SRC) specifies the first MAP register to be transferred. The next word of data (register-DES) specifies the first MAP register to which data are to be transferred.

If bit 8=0: The transfer operation is conducted as a move. The contents of n sequential MAP registers are moved, beginning from the source specified to the n sequential locations which begin at the destination register specified.

If bit 8=1: The transfer operation is conducted as a swap. The contents of n sequential MAP registers, beginning from the source register specified, are exchanged with the contents of the n sequential registers which begin with the specified destination register.

PUSH

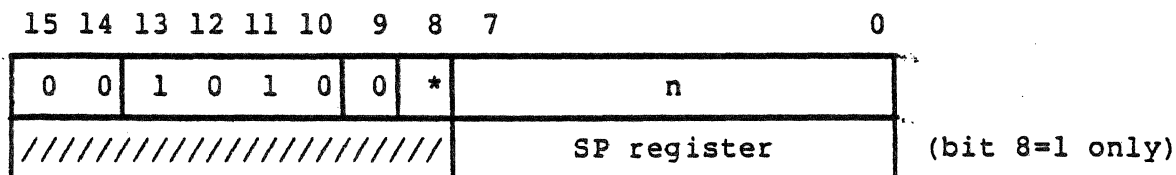


The PUSH command is used to transfer the contents of the top n words of a stack area of the 256 MAP registers onto the top of the selected stack. The operand field of the command (n) specifies the two's complement of the number of MAP registers to be transferred. After the contents of the registers have been transferred onto the stack, the MAP register selected as the stack pointer is incremented to point to the last word of data transferred onto the stack.

If bit 8=0: The MAP register selected as the stack pointer is 27--the Transformation Matrix Address (TMADR).

If bit 8=1: The next word of data input (SP register) is taken as the address of a MAP register which will be used as the stack pointer.

POP

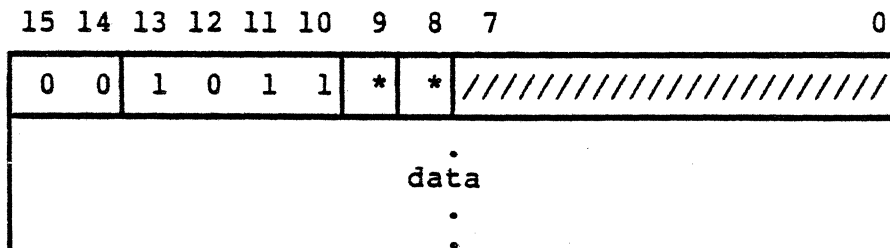


The POP command is used to decrement by n the MAP register which is functioning as a stack pointer. The operand field of the command (n) specifies the two's complement of the amount to decrement the stack pointer.

If bit 8=0: The MAP register selected as the stack pointer is 27--the Transformation Matrix Address (TMADR).

If bit 8=1: The next word of data input (SP register) will be used as the address of a MAP register to be used as the stack pointer.

MATCON



The MATCON (Matrix Concatenation) command is used to cause 16 words of data to be input which are then used as four rows of four elements to form a 4x4 matrix. This input matrix is then post-multiplied by the current Transformation Matrix (as indicated by the TMADR register) and the resultant normalized, compound matrix stored into the current Transformation Matrix.

If bit 8=0: The input matrix is post-multiplied by the Transformation Matrix, with the resultant matrix stored in the 16 words of the matrix stack immediately above the current Transformation Matrix. The 16 MAP registers above the Transformation Matrix are used to hold the intermediate or temporary results of the matrix multiplication.

If bit 8=1: The input matrix is post-multiplied by the Transformation Matrix, with the resultant matrix stored in the 16 words of the Matrix Stack immediately above the current Transformation Matrix. The TMADR is then incremented by 16, thereby pushing the previous contents of the Transformation Matrix onto the Matrix Stack and leaving the newly-concatenated compound matrix in the current Transformation Matrix.

If bit 9=0: The current Transformation Matrix is treated as a matrix whose rows contain the elements A_{ij} (i.e., A_{11} , A_{12} , A_{13} , A_{14} , A_{21} , A_{22} ,...). The resultant matrix is also treated as a matrix whose rows contain the elements A_{ij} .

If bit 9=1: The current Transformation Matrix is treated as a matrix whose rows contain the elements A_{ji} (i.e., A_{11} , A_{21} , A_{31} , A_{41} , A_{12} , A_{22} , ...) or the transpose of the typical Transformation Matrix. The resultant matrix is also

treated as a matrix whose rows contain the elements A_{ji} . This ability to treat the Transformation Matrix as the transpose allows the Picture Processor to effectively perform pre- or post-multiplication of matrices. This may be explained by the statement of some elementary axioms of matrix algebra:

- (a) $[A][B]$ not equal $[B][A]$ (in general)
- (b) $[A]' = \text{transpose of } [A]$
- (c) $([A]')' = [A]$
- (d) $[A]'[B]' = [BA]'$

In (d) above, $[B]$ is the Transformation Matrix and $[A]$ is the matrix which is input. Then: if $[A]$ is input as a transposed matrix and bit 9=1, $[B]$ will be treated as a transposed matrix resulting in a compound temporary matrix $[BA]'$ which, upon storing in the Transformation Matrix, will again be transposed. This results in the compound matrix $([BA]')'$ or $[BA]$ in the Transformation Matrix. Matrix pre-multiplication may be performed in this manner.

b. RSR Drawing Commands

The Drawing command is used to specify the type of data to be input and the manner in which the data are to be processed. The Drawing command's basic format is shown in Figure 2.3-4.

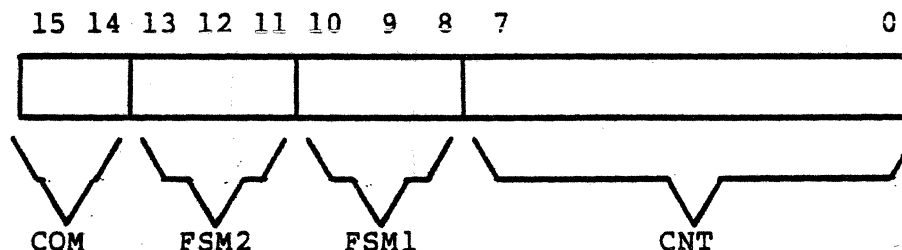


Figure 2.3-4
Basic Format of the Drawing RSR Commands

Bits of the Drawing RSR are divided into three fields:

1. Command (COM) - specifies what command is to be executed. This field is never automatically modified.
2. Finite State Machines (FSM1, FSM2) - these bits give further specification of the command to be executed. This field is automatically updated after each command execution.
3. Count (CNT) - these bits determine how many times the command should be executed before another RSR is required. This field is automatically incremented after each command execution until CNT=0; at this time, the next RSR to be executed is input.

The above fields are described in the following sections.

Command (COM)

COM=01-2DDRAW

Two words are accessed from the MAP input FIFO and are then processed by the Picture Processor as specified by the Finite State Machine bits. CNT is then incremented.

COM=10-3DDRAW

Three words are accessed from the MAP input FIFO and are then processed by the Picture Processor as specified by the Finite State Machine bits. CNT is then incremented. Note that for FSM2=5 (Pass Formatted), only two words of data will be output.

COM=11-4DDRAW

Four words are accessed from the MAP input FIFO and are then processed by the Picture Processor as specified by the Finite State Machine bits. CNT is then incremented. Note that for FSM2=5 (Pass Formatted), only two words of data will be output.

Finite State Machines (FSM1,FSM2)

FSM1- The FSM1 bits are used to describe the type of draw desired. Upon completion of the command execution, the bits are updated. The type, update definitions and FSM1 sequences initiated are listed below:

<u>FSM1 OCTAL VALUE</u>	<u>TYPE</u>	<u>VALUE AFTER UPDATE</u>	<u>SEQUENCES</u>
0	Moveto(M)	1	(M,D,M,D,...)
1	Drawto(D)	0	(D,M,D,M,...)
2	Moveto(M)	3	(M,D,D,D,...)
3	Drawto(D)	3	(D,D,D,D,...)
4	Moveto(M)	4	(M,M,M,M,...)
5	not used	4	
6	not used	7	
7	not used	6	

Interpretation of the FSM1 values follows:

Moveto: specifies a point in the coordinate system, normally used as the beginning point of a line. Note that FSM1=4 (M,M,M,...) is used to specify data which are to be processed as dots.

Drawto: indicates that a line is to be drawn from the last specified point to the point being specified.

FSM2- The FSM2 bits are used to specify whether the data accompanying DRAW commands are to be interpreted as Absolute, Relative (added to previous data), Offset (added to previously set origin) or as data which are to be Passed through the MAP, unprocessed, for output to memory or the Picture Generator. Upon completion of the command execution, the bits are updated. The interpretation, update definitions and FSM2 sequences initiated are listed below:

FSM2 OCTAL VALUE	TYPE	VALUE AFTER UPDATE	SEQUENCES
0	Set Base(SB)	1	(SB,O,O,O,...)
1	Offset(O)	1	(O,O,O,O,...)
2	Absolute(A)	3	(A,R,R,R,...)
3	Relative(R)	3	(R,R,R,R,...)
4	Absolute(A)	4	(A,A,A,A,...)
5	Pass Formatted(PF)	5	(PF,PF,PF,PF,...)
6	Pass Conditional(PC)	6	(PC,PC,PC,PC,...)
7	Pass(P)	7	(P,P,P,P,...)

Interpretation of the FSM2 values follows:

Set Base: specifies that the next set of coordinates input are to be loaded into the BASE (and INPUT) register of the MAP. For a 2DDRAW, only the x and y registers are loaded; for a 3DDRAW only, the x, y and z registers are loaded; for a 4DDRAW only, the x, y, z and w registers are loaded. For subsequent DRAW commands, the INPUT and BASE registers are used to compute the X, Y, Z and W coordinates as absolute data. A 4DDRAW Set Base should be done for 2DDRAW or 3DDRAW commands to establish the Z and W, or just W, coordinates for the input data.

Offset: specifies that subsequent data input for this command are to be processed as an offset from the current contents of the BASE register, thus treating the BASE register as an origin for subsequent data. For a 2DDRAW command, the X, Y coordinates are taken from the MAP input FIFO and the X, Y Base register values are added to them to form the X, Y input coordinates. The Z, W coordinates are taken from the BASE register. For a 3DDRAW command, the X, Y, Z coordinates are taken from the MAP input FIFO and the X, Y, Z, W Base register values are added to them to form the X, Y, Z, W input coordinates. For all DRAW commands, the BASE register is not updated.

Absolute: specifies that this set of coordinates is to be treated as absolute data. For a 2DDRAW command, the X, Y coordinates are taken from the MAP input FIFO and the Z, W coordinates are taken from the BASE register. For a 3DDRAW command, the X, Y, Z coordinates are taken from the MAP input FIFO and the W coordinate is taken from the BASE register. For a 4DDRAW command, the X, Y, Z, W coordinates are taken from the MAP input FIFO. For all DRAW commands, the new X, Y, Z, W coordinates are used to update the BASE register.

Relative: specifies that this set of coordinates is to be added to the current contents of the BASE register to form a set of absolute coordinates which are relative to the BASE register (i.e., the previous point). For all DRAW commands, the new X, Y, Z, W coordinates are used to update the BASE register.

Pass Formatted: specifies that subsequent data input for this command are to be passed through the MAP unprocessed, but formatted on output for input to the Picture Generator. For a 2DDRAW command, the X, Y coordinates are taken from the MAP input FIFO and the Z coordinate is undefined. For 3DDRAW and 4DDRAW commands, the X, Y, Z coordinates are taken from the MAP input FIFO. For all DRAW commands the output is 12 bits of X, 12 bits of Y and 6 bits of Z information in the format specified in Section 2.3.3. The BASE register is not updated by a Pass Formatted command. The FSML field is used to determine whether the word is a Moveto or Drawto command.

Pass Conditional: specifies that subsequent data input for this command are to be passed through the MAP unprocessed, but output (unformatted in any manner) only if the last point processed by the MAP was not clipped. This allows data to be output from the MAP only if the last point would be "seen". This mode is used to conditionally send characters to the Picture Generator only if the Moveto used to position the character string was also output. Use of this mode allows character strings to be "clipped" if they are currently not in the field-of-view. If data are to be output, this command functions identically to the Pass command.

Pass: specifies that subsequent data input for this command is to be passed through the MAP unprocessed, and the output unformatted in any manner. This command is used to output character strings and status commands to the Picture Generator. See Section 2.4.3 for detailed data formats for the Picture Generator.

Count (CNT): this 8-bit field is used to specify how many times the command specified by the COM bits is to be repeatedly executed (with additional data each time). The field is treated as a two's complement number from -1 (377) to -256 (000) (the sign bit is always implied). The command is executed repeatedly, with the field incremented at the end of each command execution until it goes from -1 (377) to -256 (000). At that time, the MAP will attempt to input a new RSR from the MAP input FIFO unless the RSR HOLD bit is set in the MSR. When an RSR command is input from the FIFO, it is automatically placed in the MMRSR register (SCB:177755). The MMRSR register is never directly loaded by the host computer.

2.3.3 The MAP Output Formatter

The MAP Output Formatter is a Picture System device which can function as either an Active or Passive device. In either operation, the function of the Output Formatter is to output the data directed to it by the MAP in the format required by the RSR command which was just executed. The Active/Passive state of the Output Formatter is selected by a bit in the MAP Status Register (MSR). When the MAP Output Formatter is selected as an Active device, there is an associated MAP Active Output Address register (MAOA) which holds the address of the location in PSMEM into which the next word of data output is to be stored.

- a. MAP Active Output Limit (MAOL) - SCB:177750
The MAOL is a 16-bit R/W register which holds the upper limit address within PSMEM that is to be used by the MAP Output Formatter. The Output Formatter will output data up to, but not including, the address contained in MAOL. See MAOA below.
- b. MAP Active Output Address (MAOA) - SCB:177751
The MAOA is a 16-bit R/W register. The next word of data output by the Output Formatter will be stored in the PSMEM location addressed by MAOA. This register is incremented after each word of data is stored into PSMEM. When MAOA=MAOL, the MAP Output Stopped (MOS-TOPPED) bit in the MSR is set, indicating that no more data can be output by the Output Formatter. This condition can be cleared by resetting MAOA or MAOL so that MAOA is not equal to MAOL.

If the MAP Output Formatter is not selected as an Active device, it functions as a Passive device and all data output must be taken by an Active device from its Passive Output Port (MPOP, SCB:177776).

- c. MAP Passive Output Port (MPOP) - SCB:177776
The MAP Passive Output Port is a 16-bit read-only location in the Picture System SCB from which data output by the MAP may be taken by an Active device. In order to use the MPOP, the Active device's input address register should be set to address the MPOP using the Direct I/O path to the PICTURE SYSTEM and the MAP Output Formatter selected for passive output. Once this has been completed, all subsequent data output by the MAP will be transferred to the Active device without direct intervention by the host CPU.

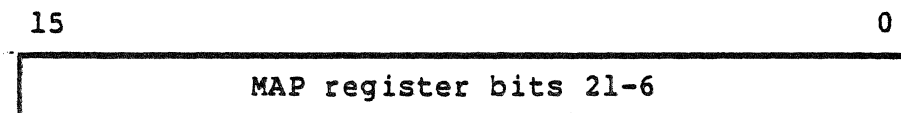
PS2/MPS Reference Manual
Hardware Details

The MAP Output Formatter may be selected to inhibit the output of any data from the MAP by setting the Inhibit Output (IOUT) bit in the MSR. When this bit is set, input to the MAP may proceed, with the data being processed as if output were not inhibited.

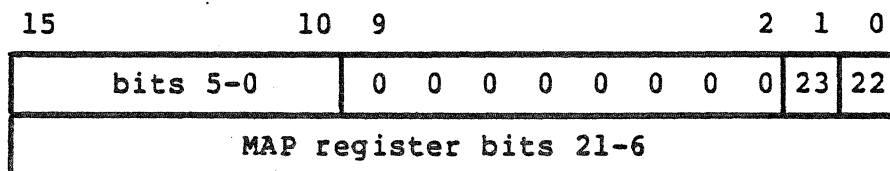
The exact data format for each word output is determined by the RSR command which was executed to produce the output from the MAP. Details of the data formats for each RSR command follows.

RSR Control Commands

The only control commands which cause data to be output from the MAP are the STORE and STORESTK commands. The output data format for these commands is identical. If the Precision bit of the RSR command (bit 8) is zero, one 16-bit word of data is output for each MAP register to be stored as shown below.



If the Precision bit of the RSR command (bit 8) is one, two 16-bit words of data are output for each MAP register to be stored as shown below.



In this extended precision mode, all 24 bits of each MAP register are output with MAP register bits 23, 22 and 5-0 are stored into bits 1,0 and 15-10 of the first word output; MAP register bits 21-6 are stored in the second word output.

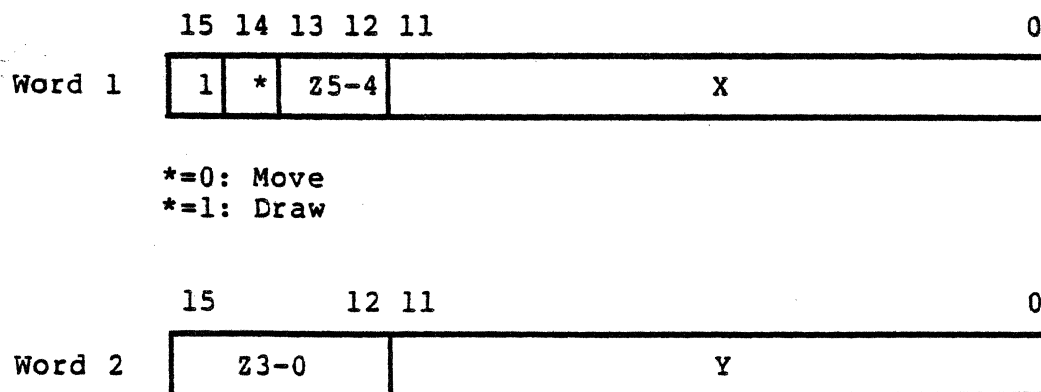
RSR 2DDRAW, 3DDRAW and 4DDRAW Commands

The output data formats for the 2DDRAW, 3DDRAW and 4DDRAW commands are identical. The 2D, 3D or 4D specification is used merely to determine the number of coordinates to be input to the MAP. Thus, the output data for these command formats may be detailed together. The output data formats for the DRAW commands are controlled by the MAP mode bits of the MSR (MMODE, bits 9,8). The status of these bits determine how the MAP is to process the two-, three- or four-dimensional data which are input and in what format the data are to be output. The output data formats for each of the different modes are detailed below.

MMODE=00 - Display Mode

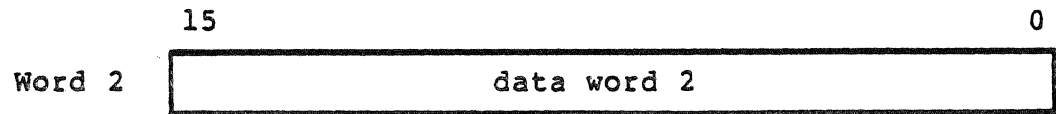
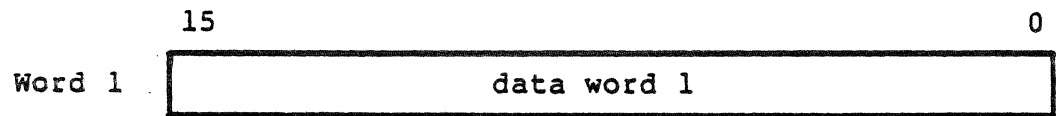
This mode causes all data output for DRAW commands to be formatted for input to the Picture Generator. The data format is:

For all DRAW commands when FSM2=0-5, two 16-bit words of data are output for each line segment endpoint output by the MAP as shown below:

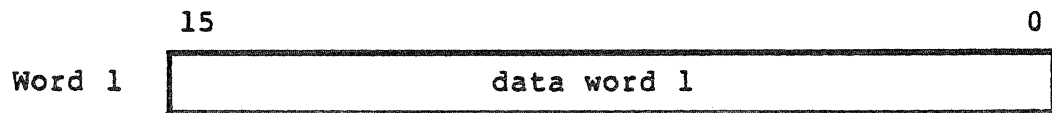


For all DRAW commands when FSM2=6, two 16-bit words of data are output for each DRAW command execution only if the last data point processed by the MAP was unclipped. The two words of data output are identical to the first two words of data input for this command as shown below. If more than two words of data are input by the MAP (i.e., for 3DDRAW or 4DDRAW commands), any extra words of data input are discarded.

PS2/MPS Reference Manual
Hardware Details



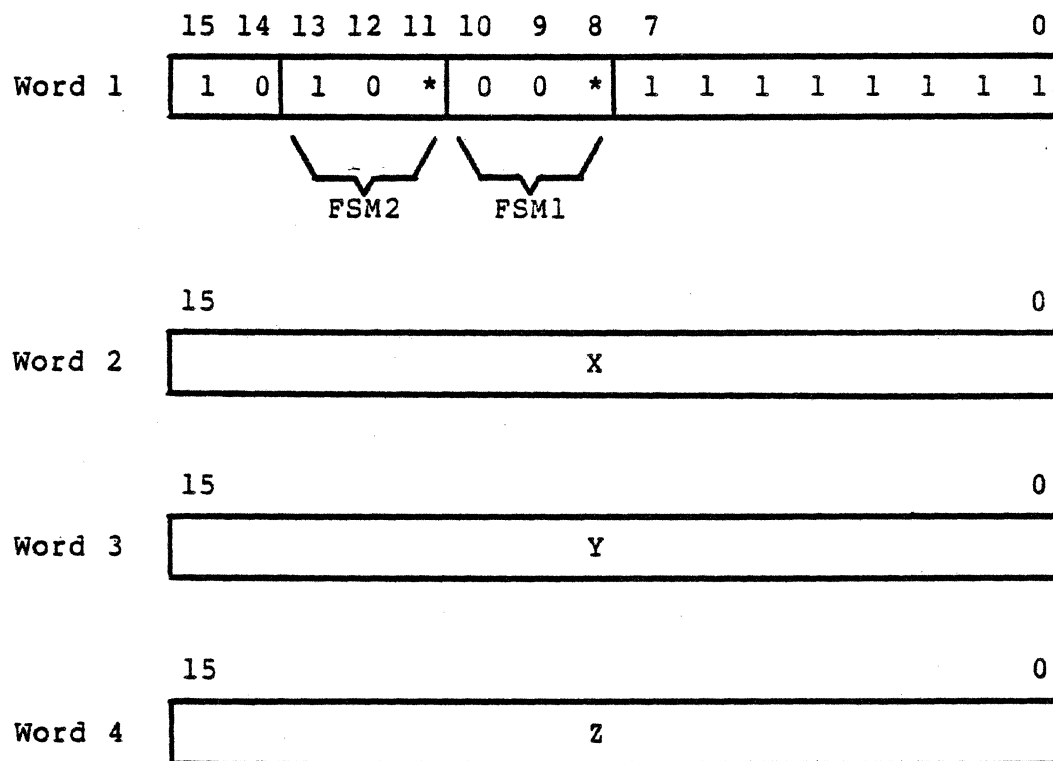
For all DRAW commands when FSM2=7, two 16-bit words of data are output for each DRAW command execution. The two words of data output are identical to the first two words of data input for this command as shown below. If more than two words of data are input by the MAP (i.e., for 3DDRAW or 4DDRAW commands), any extra words of data input are discarded.



MMODE=01 - Sixteen Bit Precision Mode

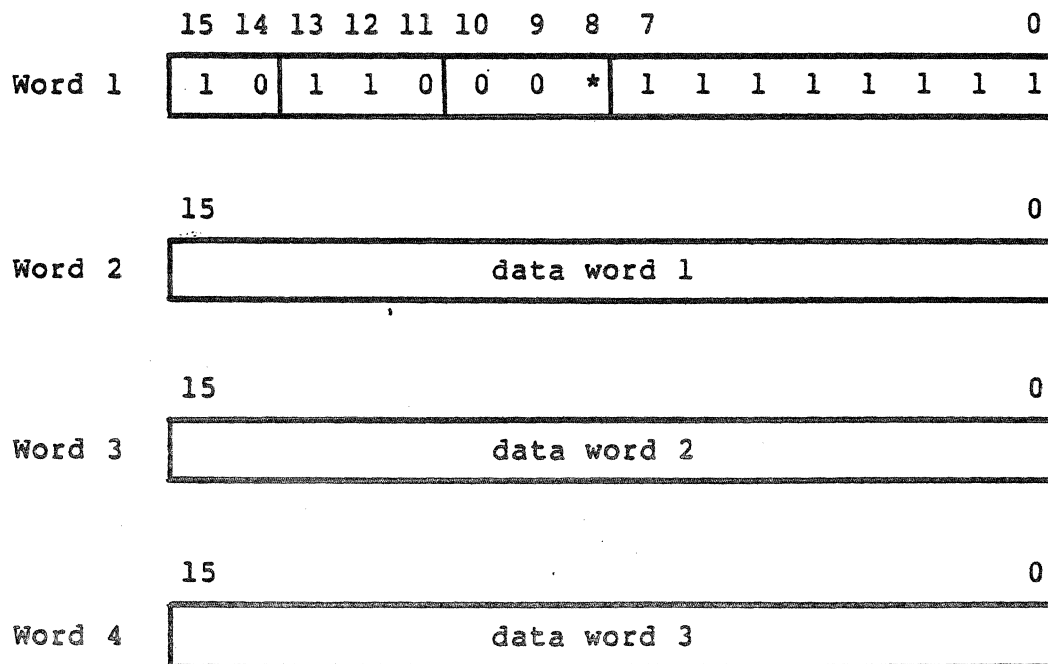
This mode causes all data output for DRAW commands to be formatted as a 3DDRAW RSR command which describes how the three words of data are to be interpreted. Four words of data are always output for each DRAW command execution. FSM1=0 for a Move and FSM1=1 for a Draw. The data format is:

For all DRAW commands when FSM2=0-5, four 16-bit words of data are output for each line segment endpoint output by the MAP. The data output will always be absolute with FSM2=4, except when in Pass Formatted mode where FSM2=5.

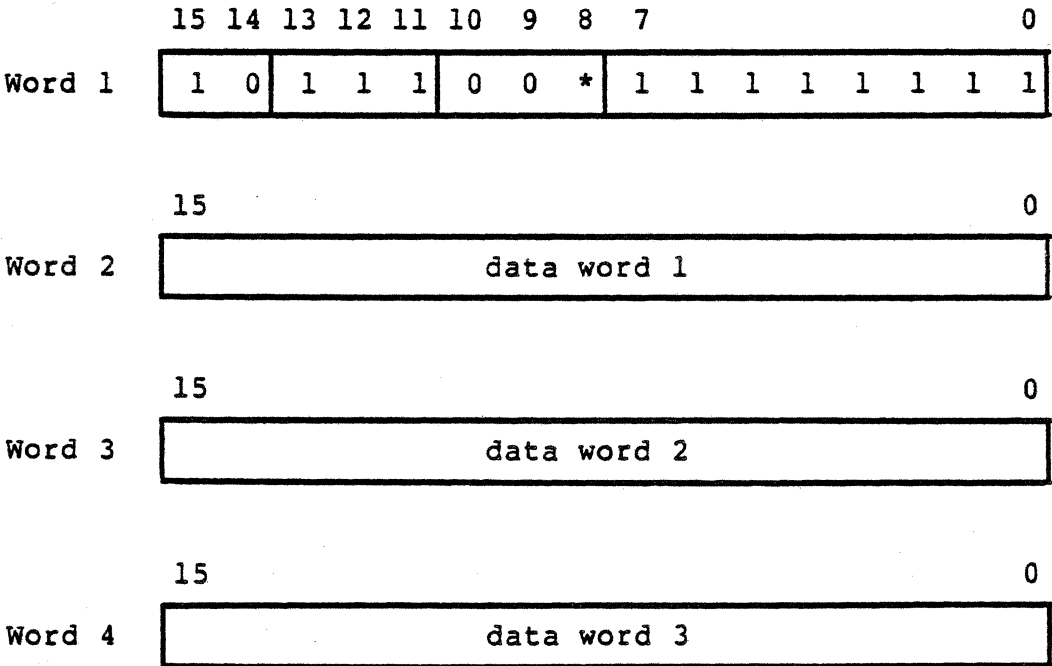


PS2/MPS Reference Manual
Hardware Details

For all DRAW commands when FSM2=6, four 16-bit words of data are output for each DRAW command execution only if the last data point processed by the MAP was unclipped. There are four words of data output; the first of which is an RSR command. The remaining words consist of the first three words of data that were input for this command as shown below. If less than three words of data were input by the MAP (i.e., for a 2DDRAW command), the fourth word of data output will be invalid data. If more than three words of data were input by the MAP (i.e., for a 4DDRAW command), the extra word of data input is discarded.



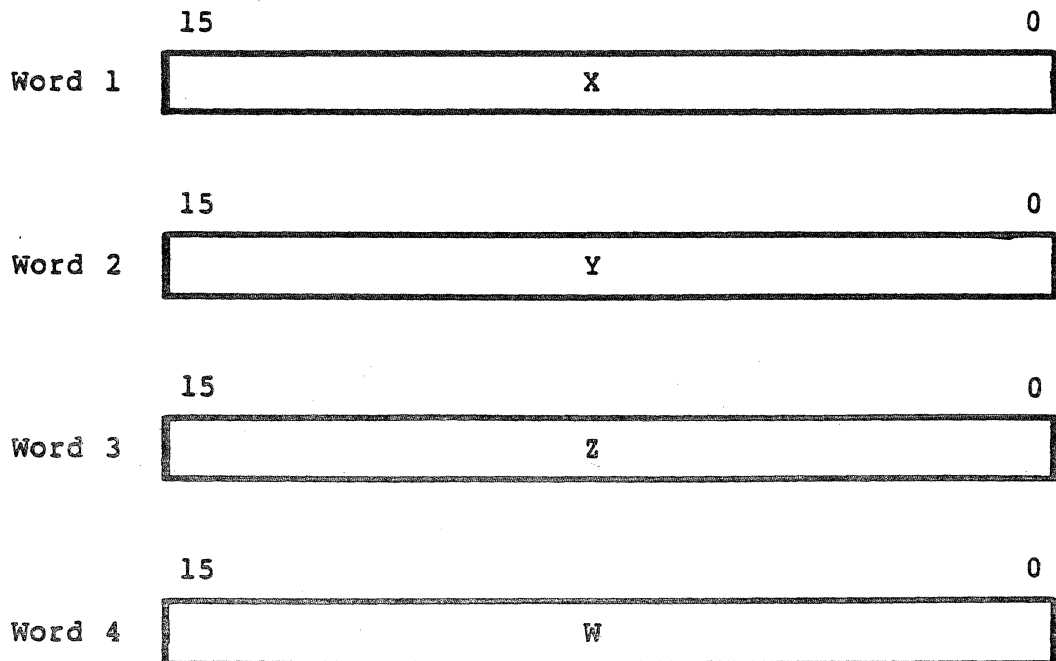
For all DRAW commands when FSM2=7, four 16-bit words of data are output for each DRAW command execution. There are four words of data output; the first of which is an RSR command. The remaining three words consist of the first three words of data that were input for this command as shown below. If less than three words were input by the MAP (i.e., for a 2DDRAW command), the fourth word of data output will be invalid data. If more than three words of data were input by the MAP (i.e., for a 4DDRAW command), any extra words of data input are discarded.



MMODE=10 - Transform/Normalize Mode

This mode causes all data output for DRAW commands to be output as four words of data which represent the transformed and normalized X, Y, Z, W coordinates of the point just processed by the MAP. The data format is:

For all DRAW commands when FSM2=0-4, four 16-bit words of data are output for each line segment endpoint input to the MAP. The output format is shown below:



For all DRAW commands when FSM2=5-7, four 16-bit words of data are output for each DRAW command execution. The four words of data output are the same four words of data that were input for this command as shown for FSM2=0-4. If less than four words of data were input by the MAP (i.e., for 2DDRAW or 3DDRAW commands), the third and fourth (or just fourth) words of data output will be invalid data.

MMODE=11 - Transform Only Mode

This mode causes all data output for DRAW commands to be output as four words of data which represent the transformed X, Y, Z, W coordinates of the point just processed by the MAP. The data format is:

For all DRAW commands when FSM2=0-4, four 16-bit words of data are output for each line segment endpoint input to the MAP. The data output consists of 1 sign bit and 15 bits of significance. It should be noted that the process of transforming the input coordinates may cause the true sign of a coordinate to be reflected by more than 1 sign bit. This is accounted for in the MAP by designating 3 bits of sign for each coordinate processed, with the result typically normalized so that 1 sign bit reflects the true sign of the number. However, when using this mode, the number is not normalized; hence, the 16-bits of data output for each coordinate may not reflect the true sign of the number. The output format is identical to that shown for MMODE=10.

For all DRAW commands when FSM2=5-7, four 16-bit words of data are output for each DRAW command execution. The four words of data output are the same four words of data that were input for this command as shown for MODE=10. If less than four words of data were input by the MAP (i.e., for 2DDRAW or 3DDRAW commands), the third and fourth (or just fourth) words of data output will be invalid data.

2.3.4 Picture Processor Maintenance Registers

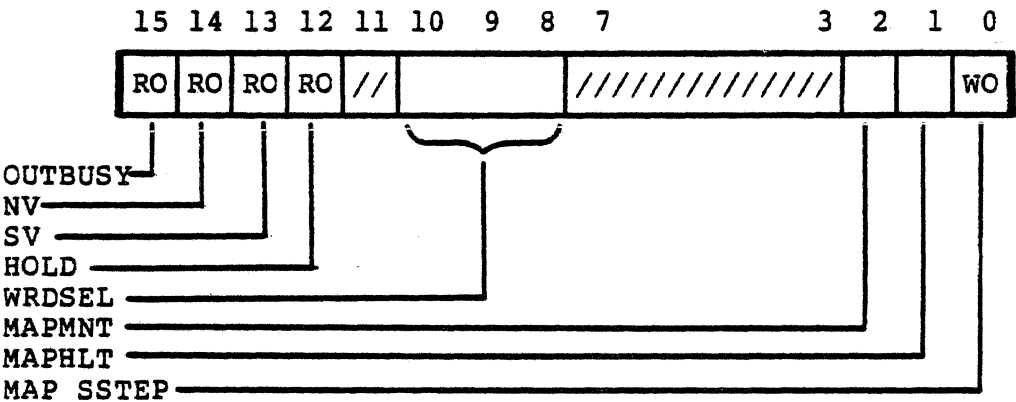
The maintenance registers of the Picture Processor provide access to the internal registers and data paths of the Picture Processor. Using these maintenance registers allows the Picture Processor microinstructions to be executed in single-step mode and also allows the microcode instruction register (or "DOIT" register) to be read and written providing complete control of the Picture Processor. Additionally, maintenance registers provide the way in which to load the Picture Processor microcode when it is equipped with the writeable control store feature. Other than the initial clearing of the MAP HLT and MAP SNGL STP bits of the MMSR register (SCB:177754, see below), these registers need not be accessed during normal operation.

The Picture Processor is controlled by eight registers in the SCB as shown in Figure 2.3-5. The maintenance registers (SCB:177754-177757) are described in detail in the following sections.

SCB:177750	Picture	<i>MAOL</i>
177751	Processor	<i>MAOA</i>
177752	Control	<i>MAIA</i>
177753	Registers	<i>MSR</i>
177754	MMSR	
177755	MMRSR	
177756	MMPAR	
177757	MMBUS	

Figure 2.3-5
Picture Processor Control and Maintenance Registers

a. MAP Maintenance Status Register (MMSR)-SCB:177754



The MMSR is used to provide status and control of the Picture Processor for maintenance purposes.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	OUTBUSY	Setting OUTBUSY indicates the MAP Output Formatter is busy. Read only. Cleared by RESET.
14	NV	NV is a status indicator which, when set, indicates that the new point has been clipped. NV, in conjunction with SV (bit 13, see below), can be used to determine how the current line has been clipped. (See SV for further details.) Read only. Cleared whenever a new point is input by the MAP for processing and RESET.
13	SV	SV is a status indicator which, when set, indicates that the saved point (the unclipped x,y,z and w coordinates of the previously processed point) has been clipped. This bit, in conjunction with NV, can be used to determine how the current line has been clipped (if at all). The bits are valid only at the end of the processing of a point when the MAP is in

the HOLD state. Read only. Cleared by RESET and whenever a new point is input by the MAP for processing.

12 HOLD

HOLD is a status indicator which, when set, indicates that the MAP is in the HOLD state awaiting another RSR command for processing. Read only. Set by the MAP Output Formatter after the last point has been processed and RESET.

11 not used

10-8 WRDSEL

WRDSEL (Control Store Word Select) is used to select which part of the 96-bit "DOIT" register can be read or written via the MMBUS register. The WRDSEL field is used to address the DOIT register in the following manner:

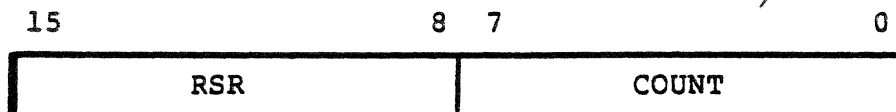
0	DOIT bits 95-80
1	DOIT bits 79-64
2	DOIT bits 63-48
3	DOIT bits 47-32
4	DOIT bits 31-16
5	DOIT bits 15-0
6	Write DOIT register to Writeable Control Store word addressed by PRMADR field of MMPAR.
7	Read Control Store word addressed by PRMADR field of MMPAR.

When the WRDSEL field overflows ($7 \Rightarrow 0$), the PRMADR field of the MMPAR register is incremented by one allowing sequential locations of the Control Store to be read or written easily. WRDSEL is used in conjunction with MAPMNT (bit 2, see below). Cleared by RESET.

7-3 not used

2	MAPMNT	When MAPMNT is set, it enables the Control Store "DOIT" register to be read or written by the MMBUS register. The WRDSEL field is used to select which bits of the "DOIT" register are to be accessed. MAPMNT must be set for the "DOIT" register to be accessed. Set by RESET. Must be programmably cleared before normal use of the Picture Processor may proceed.
1	MAPHLT	Setting MAPHLT disables the Picture Processor system clock which normally issues a clock pulse each 150 nsec. This freezes the Picture Processor for diagnostic purposes. Set by RESET. Must be cleared before normal use of the Picture Processor may proceed.
0	MAP SSTEP	This bit is set to issue a single clock pulse to the Picture Processor. Write only. Always reads as a zero.

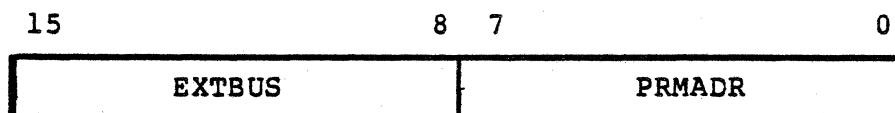
b. MAP Maintenance RSR (MMRSR)-SCB:177755 *Read only*



The MAP Maintenance RSR (Repeat Status Register) is a read-only register containing the current instruction being processed by the MAP. The fields of this register are detailed below.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-8	RSR	This field will contain the current Repeat Status Register command being executed by the MAP. (See Section 2.3.2.2 for RSR command details.) Does not clear by RESET.
7-0	COUNT	This field contains the current count of the MAP RSR instruction only when the MAP is in the HOLD state (see MMSR, bit 12). When the MAP is not in the HOLD state, this field contains the current address into the MAP RAMs. Read only. Cleared by RESET.

c. MAP Maintenance PROM Address Register (MMPAR)-SCB:177756



The MAP Maintenance PROM Address Register provides the fields which allow the 24-bit data paths to be accessed to their full precision and the Control Store to be addressed.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-8	EXTBUS	This field contains the extended bits of the MAP B-Bus (a 24-bit data path). EXTBUS contains bits 5-0 and 23,22 of the B-Bus in bits 15-8 of the MMPAR register.
7-0	PRMADR	This field contains the next state address of the Control Store MAP state machine. Read only. Cleared by RESET. When the Picture Processor is in maintenance mode (MAPMNT=1), PRMADR becomes a R/W field.

d. MAP Maintenance B-Bus (MMBUS)-SCB:177757

This register contains bits 21-6 of the main internal B-Bus of the MAP. MMBUS, in conjunction with EXTBUS, may be used to read the 24-bit B-Bus. When the Picture Processor is in maintenance mode (MAPMNT=1), the "DOIT" register may be read or written by reading or writing MMBUS (WRDSEL selects which 16 bits of the DOIT register are to be accessed). Read/Write. Cleared by RESET. Any access to this register will increment the WRDSEL field of the MMSR after the operation.

The following table details the data which may be read or written using this register for each of the WRDSEL values. The DOIT bit <95-0> and the name of the particular function performed for each bit are also included. If the content of a bit field is not specified, it should be considered undefined.

PS2/MPS Reference Manual
Hardware Details

<u>WRDSEL</u>	<u>MMBUS Register Contents</u>
0	MMBUS<15>=DOIT<95> NBADR(7) MMBUS<14>=DOIT<94> NBADR(6) MMBUS<13>=DOIT<93> NBADR(5) MMBUS<12>=DOIT<92> NBADR(4) MMBUS<11>=DOIT<91> NBADR(3) MMBUS<10>=DOIT<90> NBADR(2) MMBUS<09>=DOIT<89> NBADR(1) MMBUS<08>=DOIT<88> NBADR(0) MMBUS<07>=DOIT<87> *LDSUB MMBUS<06>=DOIT<86> *SELSUBRTN MMBUS<05>=DOIT<85> *RSRUPDAT MMBUS<04>=DOIT<84> *RSRLOAD MMBUS<03>=DOIT<83> *GATFIFO MMBUS<02>=DOIT<82> *OUTPUTSET MMBUS<01>=DOIT<81> *OUTPUTLOAD MMBUS<00>=DOIT<80> *SETPNT
1	MMBUS<15>=DOIT<79> *RAMWRT MMBUS<14>=DOIT<78> *MBSEL MMBUS<13>=DOIT<77> *COUNTSEL MMBUS<12>=DOIT<76> *DTREGA MMBUS<11>=DOIT<75> *DTREGB MMBUS<10>=DOIT<74> *DTREGOUT MMBUS<09>=DOIT<73> *MMPSELG MMBUS<08>=DOIT<72> *SELSH MMBUS<07>=DOIT<71> *MNTOUT MMBUS<06>=DOIT<70> *MBCBSEL MMBUS<05>=DOIT<69> *MDACBSEL MMBUS<04>=DOIT<68> *RACBSEL MMBUS<03>=DOIT<67> *MBLOAD MMBUS<02>=DOIT<66> *MDALOAD MMBUS<01>=DOIT<65> *RALOAD MMBUS<00>=DOIT<64> *LOGICF
2	MMBUS<15>=DOIT<63> *CARRIN MMBUS<14>=DOIT<62> *FUNSEL0 MMBUS<13>=DOIT<61> *FUNSEL1 MMBUS<12>=DOIT<60> *FUNSEL2 MMBUS<11>=DOIT<59> *FUNSEL3 MMBUS<10>=DOIT<58> *MRALOAD MMBUS<09>=DOIT<57> *NMROUT MMBUS<08>=DOIT<56> *MRASEL MMBUS<07>=DOIT<55> REG 7 MMBUS<06>=DOIT<54> REG 6 MMBUS<05>=DOIT<53> REG 5 MMBUS<04>=DOIT<52> REG 4 MMBUS<03>=DOIT<51> REG 3 MMBUS<02>=DOIT<50> REG 2 MMBUS<01>=DOIT<49> REG 1 MMBUS<00>=DOIT<48> REG 0

3

```

MMBUS<15>=DOIT<47> *NSRLOAD
MMBUS<14>=DOIT<46> *NSRMINLD
MMBUS<13>=DOIT<45> *BUSADRSEL
MMBUS<12>=DOIT<44> *CNTSELA
MMBUS<11>=DOIT<43> *CNTSELE
MMBUS<10>=DOIT<42> *CHECKSTATUS
MMBUS<09>=DOIT<41> *DISPATCH
MMBUS<08>=DOIT<40> *TRANPOSE
MMBUS<07>=DOIT<39> *AC0LOAD
MMBUS<06>=DOIT<38> *AC0INC
MMBUS<05>=DOIT<37> *AC1LOAD
MMBUS<04>=DOIT<36> *AC1INC
MMBUS<03>=DOIT<35> *AC2LOAD
MMBUS<02>=DOIT<34> *AC2INC
MMBUS<01>=DOIT<33> *AC3LOAD
MMBUS<00>=DOIT<32> *AC3INC

```

4

```

MMBUS<15>=DOIT<31> *R1ROMOUT
MMBUS<14>=DOIT<30> *R2ROMOUT
MMBUS<13>=DOIT<29> *LD_FLAGS
MMBUS<12>=DOIT<28> *USENORM
MMBUS<11>=DOIT<27> *J1B
MMBUS<10>=DOIT<26> *K1B
MMBUS<09>=DOIT<25> *USEJ1B
MMBUS<08>=DOIT<24> *USEK1B
MMBUS<07>=DOIT<23> *USESV
MMBUS<06>=DOIT<22> *USENV
MMBUS<05>=DOIT<21> *JV
MMBUS<04>=DOIT<20> *CLRV
MMBUS<03>=DOIT<19> *D
MMBUS<02>=DOIT<18> *C
MMBUS<01>=DOIT<17> *B
MMBUS<00>=DOIT<16> *A

```

5

```

MMBUS<15>=DOIT<15>
MMBUS<14>=DOIT<14>
MMBUS<13>=DOIT<13>
MMBUS<12>=DOIT<12>
MMBUS<11>=DOIT<11>
MMBUS<10>=DOIT<10>
MMBUS<09>=DOIT<09>
MMBUS<08>=DOIT<08>
MMBUS<07>=DOIT<07> *GATCOUNT
MMBUS<06>=DOIT<06> *EXTSEL
MMBUS<05>=DOIT<05> *HLTINT
MMBUS<04>=DOIT<04> *GATMRI
MMBUS<03>=DOIT<03> *LDMRI
MMBUS<02>=DOIT<02> *LOADEXT
MMBUS<01>=DOIT<01> *CLREXT
MMBUS<00>=DOIT<00> *JHIT

```

2.4 The Picture Generator

The Picture Generator is the unit of the Picture System that controls the drawing of lines and characters on the Picture Display. There are four basic units of the Picture Generator:

1. Line Generator Input Controller
2. Refresh Controller/Refresh and Device Processor
3. Line Generator
4. Character Generator

These units function together to input data from the refresh buffer area of Picture System memory (or to accept data directly from an Active device--the MAP Output Formatter, for example) and to convert the data into the appropriate analog signals to drive the CRT(s).

The Picture Generator input and status is controlled by fourteen registers in the SCB. These registers, shown in Figure 2.4-1, together with the four basic units of the Picture Generator, are described in detail in the following sections.

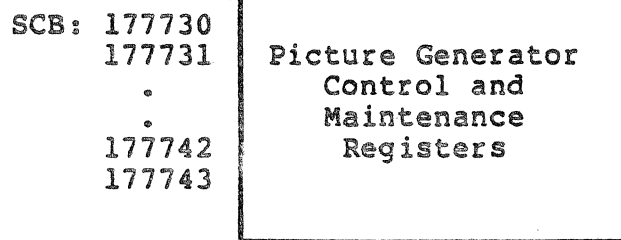


Figure 2.4-1
Picture Generator Input Control, Status and Maintenance Registers

2.4.1 The Line Generator Input Controller

The Line Generator Input Controller is a Passive device whose function is simply to transfer 16-bit words of data passively received to a 8-word Line Generator FIFO (First In First Out) buffer. As the FIFO becomes full, the Input Controller waits for the Line Generator to empty two 16-bit words from the FIFO before the next word of data is transferred to the FIFO.

a. Line Generator Passive Input Port (LGPIP)-SCB:177775

The Line Generator Passive Input Port is a 16-bit write-only location in the Picture System SCB to which data may be directed by an Active device for input to the Picture Generator. In order to use the LGPIP, the active device's output address register should be set to address the LGPIP using the Direct I/O path to the Picture System. Once this has been done, all subsequent data output by the Active device will be transferred to the Line Generator without direct intervention by the host CPU.

2.4.2 Refresh Controller/Refresh and Device Processor

The Refresh Controller and Refresh and Device Processor perform the same basic functions within the Picture System 2 and the Multi Picture System. Each controls the refreshing of images by reading data from the Picture Memory and channeling the data to the Line Generator for display. The Refresh Controller and the Refresh Processor are mutually exclusive devices on the PSBUS. Both the Refresh Controller and the Refresh Processor are assigned a set of registers in the SCB at conflicting addresses (Refresh Controller SCB:177730-177737, Refresh Processor SCB:177730-177735). Thus there may be only one of these devices within the Picture System. Which device is installed determines the type of system - Picture System 2 or Multi Picture System. Although both the Refresh Controller and Refresh Processor perform the same basic function, the two devices are not software compatible. The registers within the SCB are radically different and the Refresh Processor is capable of performing many more functions than the Refresh Controller. Each of these devices is detailed in the following sections.

2.4.2.1 Refresh Controller

The Refresh Controller is an Active device whose function is to control the refreshing of pictures on the Picture Display. The Refresh Controller reads data from the area of Picture System memory designated as refresh buffer area and transfers the data to the Line Generator Input Controller for subsequent display by the Line Generator. The Refresh Controller is also equipped with special-purpose segmentation registers to facilitate refresh buffer segmentation.

Refresh and segmentation is controlled by eight registers in the SCB. These registers, shown in Figure 2.4-2, are described in detail below:

SCB: 177730	RFCSN
177731	RFSN
177732	RFAWA
177733	RFAWL
177734	RFAIA
177735	RFASA
177736	RFAIL
177737	RFSR

Figure 2.4-2
Refresh Status and Control Registers

a. Refresh Current Segment Name (RFCSN)-SCB:177730

The RFCSN is a 16-bit R/W register which holds the name of the segment of the refresh buffer currently being refreshed by the Picture Generator. Whenever a Segment Name is encountered by the Refresh Controller, that name is loaded into the RFCSN. If RFCSN=RFSN, a segment match is indicated by setting the MATCH REQ bit in the System Interrupt Request register. If the MATCH HOLD bit in the RFSR is set, the Refresh Controller will then wait for the MATCH REQ bit to be reset before the refresh cycle continues.

b. Refresh Segment Name (RFSN)-SCB:177731

The RFSN is a 16-bit R/W register which holds the name of a segment which may be searched for during the traversing of the refresh buffer by the Refresh Controller while refreshing the display file. The search mode is enabled by setting the SEARCH bit in the RFSR. A search may be done in any of the modes available, as specified by the RFSR SEARCH MODE bits. When a search match occurs (RFCSN=RFSN), the MATCH REQ bit in the System Interrupt Request register is set, indicating that a match has been found. Note: the match need not be equal, depending upon the status of the SEARCH MODE bits in the RFSR.

c. Refresh Active Writeback Address (RFAWA)-SCB:177732

The RFAWA is a 16-bit R/W register, and is used in the "writeback" operation. When the Input Controller is in writeback mode, it reads, from the PSMEM, the word of data addressed by RFAIA, and writes it back into the location indicated in the RFAWA. Writeback mode is selected by setting the WRITEBACK bit in the RFSR and is used to automatically reclaim the refresh buffer area allocated to deleted segments. This register is automatically incremented after each word is written. When RFAWA=RFAWL, the RFHOLD bit in the RFSR is set, indicating that writeback cannot proceed. This condition may be cleared by resetting the WRITEBACK bit in the RFSR or resetting RFAWA or RFAWL so that RFAWA is not equal to RFAWL. The WBSTOP REQ bit in the System Interrupt Request Register must then be acknowledged.

d. Refresh Active Writeback Limit (RFAWL)-SCB:177733

The RFAWL is a 16-bit R/W register which holds the upper address limit within PSMEM that is to be written by the Refresh Controller during writeback. The Refresh Controller will write data up to, but not including, the address contained in RFAWL.

e. Refresh Active Input Address (RFAIA)-SCB:177734

Permanently
The RFAIA is a 16-bit R/W register which holds the PSMEM address used to fetch the next word of data for input to the Picture Generator. After each word of data is transferred to the Line Generator Input Controller, this register is incremented so that the next word of data is taken from the next sequential location of Picture System memory. When RFAIA=RFAIL, the RFSTOPPED bit in the Refresh Status Register (RFSR, see below) is set, indicating that no more data can be input by the Refresh Controller. This condition may be cleared by setting the RFSTART bit in the RFSR which then loads the RFAIA with the contents of the Refresh Active Start Address register, RFASA. This register should not be used to address any Passive device other than Picture System memory. Incrementation of this register cannot be inhibited by addressing one of the eight passive I/O ports (SCB:177770-177777).

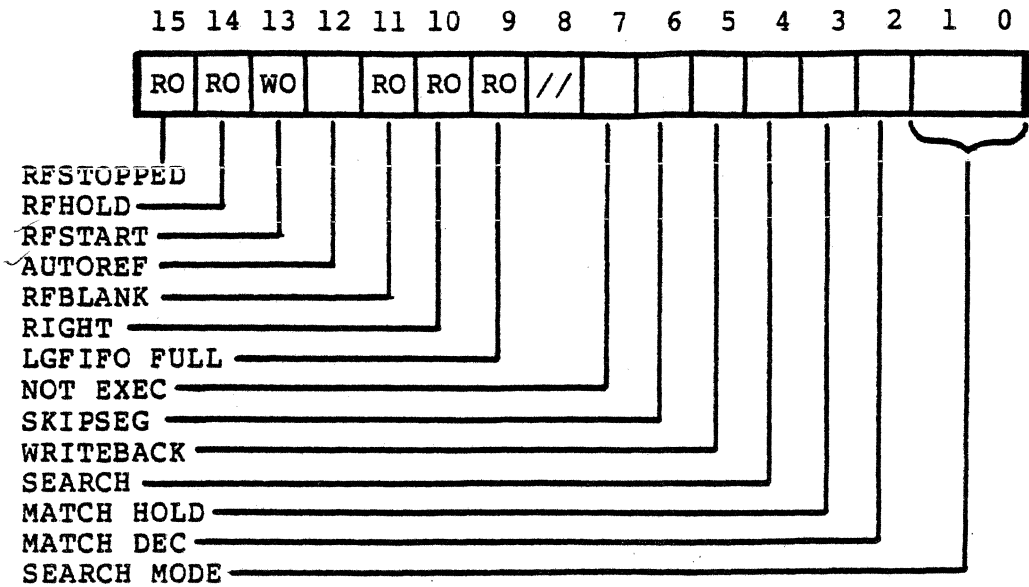
f. Refresh Active Start Address (RFASA)-SCB:177735

this register stays in sync with RFAIA
The RFASA is a 16-bit R/W register which holds the initial PSMEM address which is to be loaded into the RFAIA upon the start of a refresh cycle. The contents of this register is loaded into the RFAIA by the Refresh Controller whenever the RFSTART bit in the RFSR is set or by the Refresh Controller when an automatic refresh cycle is begun (see RFSR, AUTOREF).

g. Refresh Active Input Limit (RFAIL)-SCB:177736

The RFAIL is a 16-bit R/W register which holds the upper limit address within PSMEM that is to be used by the Refresh Controller. The Refresh Controller will input data up to, but not including, the address contained in RFAIL.

h. Refresh Status Register (RFSR)-SCB:177737



The RFSR is a 16-bit R/W register which is used to provide operating mode information to the Refresh Controller and status of the Refresh Controller back to the control program.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	RFSTOPPED	Setting RFSTOPPED (Refresh Stopped), indicates that the Refresh Controller has stopped reading data from the refresh buffer and is waiting for another refresh cycle to begin. Read only. Set by RESET and cleared by setting RFSTART.
14	RFHOLD	Setting RFHOLD (Refresh Hold), indicates that the Refresh Controller has stopped reading data from the refresh buffer since a MATCH REQ or the writeback limit has been met (RFAWA=RFAWL). Read only. Cleared by RESET or acknowledgement of MATCH REQ or WBSTOP REQ.

- 13 RFSTART This bit is set to initiate execution of a refresh cycle by the Refresh Controller. Setting this bit causes the contents of the RFASA register to be loaded into the RFAIA. All subsequent data read by the Refresh Controller will be input to the Picture Generator for display until RFAIA=RFAIL or a Refresh Status Halt is encountered. Setting this bit also clears the MATCH REQ, WBSTOP REQ and RFSTOP REQ bits. Write only. Always reads as a zero.
- 12 AUTOREF This bit is set to place the Refresh Controller into a self-operating mode of control where refreshing will proceed, without software control, at the interval of the line frequency specified by the Real Time Clock Counts register (SCB:177745). The Refresh Controller reads the contents of PSMEM from the location specified by the RFASA register to the location specified by the RFAIL register, or until a Refresh Control command with the HALT bit set (see Section 2.4.3c) is encountered. If the Refresh Controller has not completed an automatic refresh cycle when the clock interval elapses, the next refresh cycle will not begin until the clock interval again elapses. See Section 2.6.1 for further details on refresh interval selection. Cleared by RESET.
- 11 RFBLANK When RFBLANK is set, it indicates that the Refresh Controller is in the mode of operation where none of the data words read from Picture Memory are being passed to the Picture Generator for display. This mode of operation is en-
- refresh active status add*
refresh active input limit

		tered when the SKIPSEG bit (see below) is set or a Refresh Control Command has been encountered with the BLANK bit set. Read only.
10	RIGHT	This bit is a status indicator which is used to determine whether the next 16-bit word to be sent to the Picture Generator is the right or left (i.e., word 1 or word 2) part of the 32-bit Line Generator Command (see Section 2.4.3). Read only. Cleared by RESET.
9	LGIFIFO FULL	This bit is a status indicator which becomes set whenever the Line Generator Input FIFO is full. Read only. Cleared when the LG Input FIFO is not full and by PSRESET or PGRESET.
8	not used	
7	NOT EXEC	Setting NOT EXEC causes the Refresh Controller to function in a mode where the data words read from the refresh buffer are not interpreted as Line Generator commands. When used in conjunction with SKIPSEG (bit 6), this allows data to be transferred in blocks within PSMEM.
6	SKIPSEG	Setting SKIPSEG (Skip Segment), causes the Refresh Controller to skip all further data in the refresh buffer until RFSTOPPED or RFHOLD is set. This bit, in conjunction with the MATCH HOLD and SEARCH bits, allows segments to be skipped or blanked during refresh. Cleared by RESET.
5	WRITEBACK	When WRITEBACK is set, each word of data read by the Refresh Controller will be input to the Line Generator Input Controller as usual and also

written back into PSMEM in the location specified by the RFAWA register. After each word is written, the RFAWA register is incremented allowing sequential PSMEM locations to be written. When RFAWA=RFAWL, the Refresh Controller is placed into the HOLD state (with RFHOLD set) until the condition is acknowledged and cleared.

4 SEARCH

When SEARCH is set, the Refresh Controller compares all segment names encountered during a refresh cycle with the name contained in RFSN. If a match occurs (as determined by the SEARCH MODE bits, see below), the MATCH REQ bit is set in the System Interrupt Request register. If the MATCH HOLD bit is set, the Refresh Controller will halt with the RFHOLD bit set.

3 MATCH HOLD

When MATCH HOLD is set, the Refresh Controller will terminate input to the Line Generator Input Controller whenever a MATCH REQ is issued and SEARCH (bit 4) is set. The RFAIA will be addressing the PSMEM location following the segment name that was searched for. Cleared by RESET.

2 MATCH DEC

When MATCH DEC is set, the Refresh Controller, when in a SEARCH operation with the MATCH HOLD bit set, will automatically reposition the RFAIA to point at the segment name encountered rather than after it, as is the normal mode of operation when MATCH DEC = 0. This causes the RFAIA to be decremented by two on a segmented name match. Cleared by RESET.

1,0	SEARCH MODE	These bits determine how the Segment Name in the RFSN is to be used in searching for a valid match. Set to 00 by RESET. The bit combinations and interpretations are as follows:
00		Search for any name match. In this mode, any segment name encountered will cause the MATCH REQ to be issued.
01		Search for right match. In this mode, any segment name encountered whose eight least-significant bits are equal to the eight least-significant bits of the RFSN register will cause a MATCH REQ to be issued.
10		Search for left match. In this mode, any segment name encountered whose eight most-significant bits are equal to the eight most-significant bits of the RFSN register will cause a MATCH REQ to be issued.
11		Search for exact match. In this mode, any segment name encountered which is equal to the segment name in the RFSN register will cause a MATCH REQ to be issued.

2.4.2.2 Refresh and Device Processor

The Refresh and Device Processor (Refresh Processor) is an Active device containing a high-speed digital processor that can be used in a multi-user environment to control the refresh of pictures on the Picture Display. Refresh buffer segmentation is implemented with an efficient algorithm so that the Picture Controller is relieved from the task of controlling the movement of data in refresh memory for garbage collection. The Refresh Processor also relieves the Picture Controller of much of the burden associated with controlling Picture System Interactive Devices such as Light Pens, Tablets, Joysticks, Keyboards, Function Switches or Control Dials.

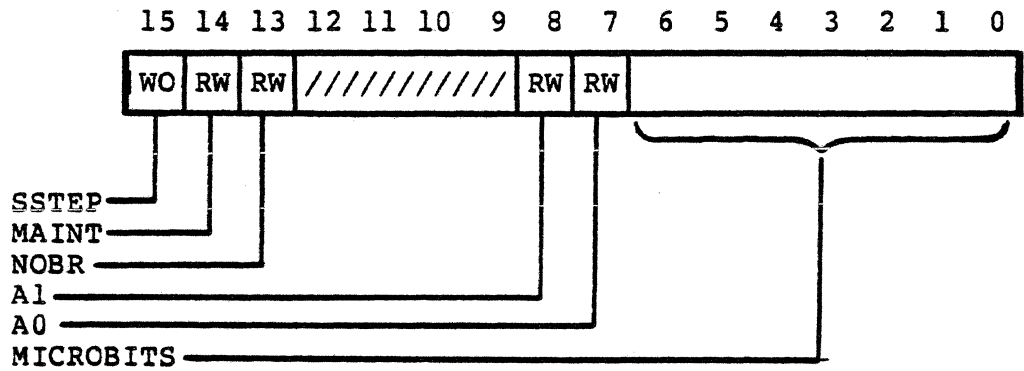
Refresh, segmentation and device functions are controlled by six registers in the SCB. These registers, shown in Figure 2.4-3, are described in detail below:

SCB: 177730
177731
177732
177733
177734
177735

RFMNT0
RFMNT1
RFPC
RFDT
RFCLK
RFEXT

Figure 2.4-3
Refresh Processor Control and Status Registers

a. Refresh Maintenance Register 0 (RFMNT0) - SCB:177730



The RFMNT0 is a 16-bit R/W register which is used to provide diagnostic capabilities to the Refresh Processor.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	SSTEP	Writing a 1 into this bit when bit 14 is set issues a clock pulse to the Refresh Processor. If this pulse causes the Refresh Processor to initiate a read or write on the PSBUS, the Refresh Processor will continue to run until that transaction is completed. Write only, read state is indeterminate.
14	MAINT	When this bit is clear, the Refresh Processor runs normally. When this bit is set, the Refresh Processor halts. Cleared by RESET.
13	NOBR	This is the No Branch bit. When it is set, the program counter in the Refresh Processor increments on every clock tick (i.e., doesn't branch). This bit, in conjunction with bits 14 and 15, is used to examine the Refresh Processor microcode. Cleared by RESET.

PS2/MPS Reference Manual
Hardware Details

12-9 not used

8 A1 This bit, in conjunction with A0 (below), is used to select which part of the 54-bit current Refresh Processor micro-word can be read from the MICROBITS field of the RFMNT0 and RFMNT1 registers.

7 A0 This bit, in conjunction with A1 (above), is used to select which part of the 54-bit current Refresh Processor micro-word can be read from the MICROBITS field of the RFMNT0 and RFMNT1 registers. The microbits selected by A1 and A0 are shown below:

A1,A0=00

RFMNT0<6>=READPS
RFMNT0<5>=WRITEPS
RFMNT0<4>=I4
RFMNT0<3>=B2
RFMNT0<2>=B1
RFMNT0<1>=A0
RFMNT0<0>=D10
RFMNT1<6>=TST1
RFMNT1<5>=TST0
RFMNT1<4>=PSA
RFMNT1<3>=DB
RFMNT1<2>=I7
RFMNT1<1>=*ALUTOBUS
RFMNT1<0>=D5

A1,A0=01

RFMNT0<6>=ALUBUFSTE
RFMNT0<5>=FIFO
RFMNT0<4>=D14
RFMNT0<3>=I3
RFMNT0<2>=D2
RFMNT0<1>=I6
RFMNT0<0>=I0
RFMNT1<6>=TST3
RFMNT1<5>=TST2
RFMNT1<4>=PUSH
RFMNT1<3>=*LATCH STATUS
RFMNT1<2>=D7
RFMNT1<1>=D6
RFMNT1<0>=*FE

A1,A0=10

RFMNT0<6>=FSEL1
RFMNT0<5>=FSEL0
RFMNT0<4>=D9
RFMNT0<3>=I8
RFMNT0<2>=A1

A1,A0=11

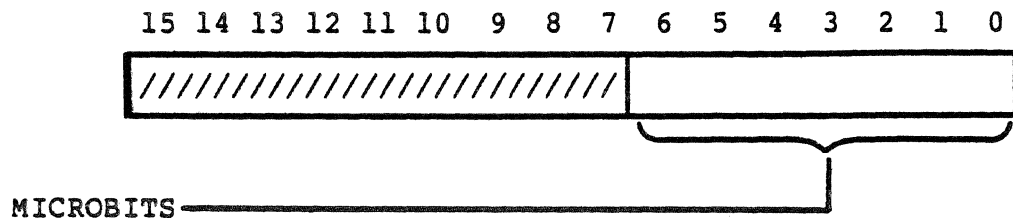
6-0

MICROBITS

RFMNT0<1>=D1
RFMNT0<0>=I5
RFMNT1<6>=PHOLD
RFMNT1<5>=INV
RFMNT1<4>=D4
RFMNT1<3>=D13
RFMNT1<2>=*PSTOBUS
RFMNT1<1>=D11
RFMNT1<0>=C0
RFMNT0<6>=FDAT
RFMNT0<5>=FSEL2
RFMNT0<4>=B3
RFMNT0<3>=D3
RFMNT0<2>=I2
RFMNT0<1>=B0
RFMNT0<0>=D0
RFMNT1<6>=REQMEM
RFMNT1<5>=GNTHOLD
RFMNT1<4>=A3
RFMNT1<3>=A2
RFMNT1<2>=D12
RFMNT1<1>=I1
RFMNT1<0>=D15

These bits contain the micro-bits currently selected by the A1 and A0 bits as described above. Read only.

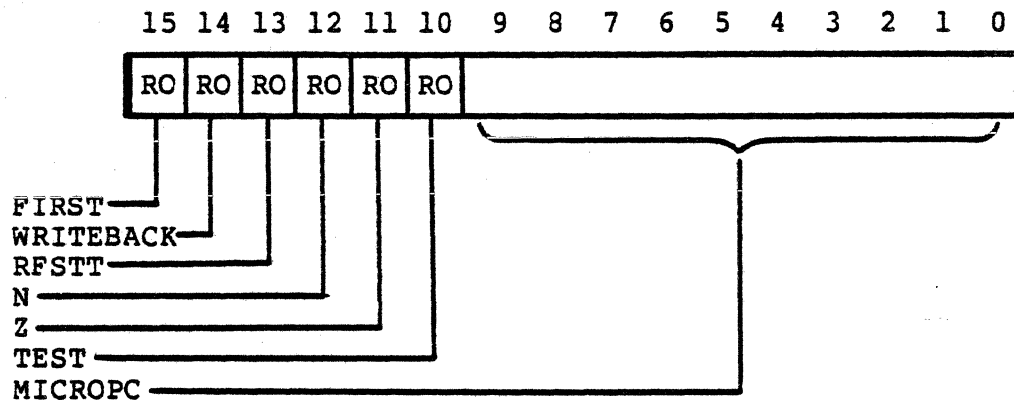
b. Refresh Maintenance Register 1 (RFMNT1) - SCB:177731



The RFMNT1 is a 16-bit read only register which is used to provide diagnostic capabilities for the Refresh Processor.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-7	not used	
6-0	MICROBITS	These bits contain the micro-bits currently selected by the A1 and A0 bits of RFMNT0. Read only.

c. Refresh Program Counter (RFPC) - SCB:177732



The RFPC is a 16-bit read only register containing the micro-controller status bits and program counter. The register fields listed below assume the contents of this register was complemented after being read.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	FIRST	This bit is set/cleared by the Refresh Processor to flag various conditions (e.g., the first occurrence of a segment name in the transformed display list).
14	WRITEBACK	This bit is set by the Refresh Processor to indicate that memory write back during garbage collection is in process, cleared otherwise.
13	RFSTT	When this bit is set it indicates that the Refresh Processor has encountered the first word of a Multi Picture System Refresh Control Command.
12	N	This bit is set to indicate that the last latched operation of the Refresh Processor arithmetic logic unit resulted in a negative number.

11	Z	This bit is set to indicate that the last latched operation of the Refresh Processor arithmetic logic unit resulted in zero.
10	TEST	This bit is set to indicate the condition that the Refresh Processor is presently testing. This bit in conjunction with the microbit INV causes the Refresh Processor to execute either the next instruction in its microcode or to branch to a new place in its microcode and execute from there.
9-0	MICROPC	These bits are the microcode program counter of the micro-controller. These bits are of an inverted sense and should be complemented after being read to determine the MICROPC.

d. Refresh Processor Data Register (RFDT) - SCB:177733

This is a 16-bit register that is used by the Refresh Processor to communicate with the rest of the Picture System. After RESET this register contains the ECO level of the Refresh Processor Hardware. The Picture Controller software should write the address of the Refresh Table (See Section 2.4.2.2.1 a) into RFDT, acknowledge the Refresh Processor System Interrupt Request bit, then set the Refresh Processor System Interrupt Enable bit in the System Interrupt Enable Register (see Section 2.5.2).²⁻²²⁵ The Refresh Processor will use this address to locate the Refresh Table and immediately begin the display and device control functions. If the software writes a 17777 into RFDT (which is not a valid address for the Refresh Table), the Refresh Processor will enter maintenance mode. The maintenance mode functions are described in Section 2.4.2.2.3. The Refresh Processor also uses this register as a port to the PSBUS, thus enabling it to read and write Picture Memory and the System Control Block (SCB). Therefore,

PS2/MPS Reference Manual
Hardware Details

under normal conditions, this register should not be accessed after the address of the Refresh Table is written into it.

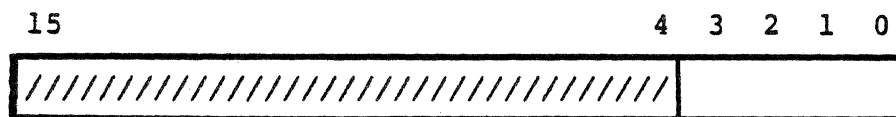
177733

15

0

STARTING ADDRESS

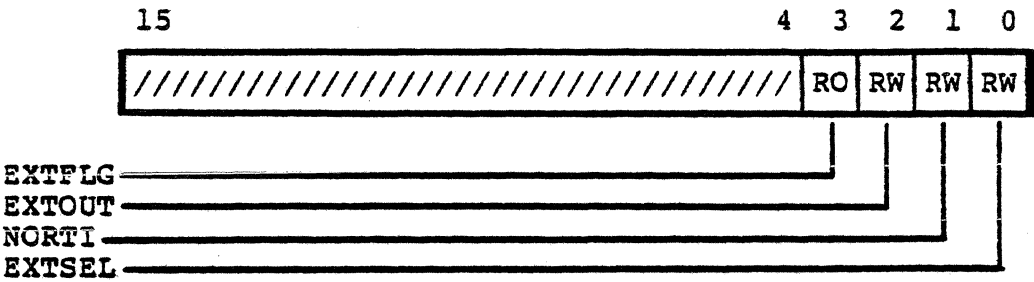
e. Refresh Processor Clock Register (RFCLK) - SCB:177734



The Refresh Processor synchronizes itself to the A.C. power source when the EXTSEL bit in the RFEXT register is clear. It refreshes the display at the frequency specified by bits 3-0 in RFCLK. The following table lists the refresh frequencies associated with the values of these four bits. This register is set to 1110 (60 Hz.) by RESET.

<u>Bit Value</u>	<u>60 Hz A.C. Power</u> <u>Refresh Frequency</u>	<u>50 Hz A.C. Power</u> <u>Refresh Frequency</u>
1111	120 Hz.	100 Hz.
1110	60 Hz.	50 Hz.
1101	40 Hz.	33.33 Hz.
1100	30 Hz.	25 Hz.
1011	24 Hz.	20 Hz.
1010	20 Hz.	16.66 Hz.
1001	17 Hz.	14.28 Hz.
1000	15 Hz.	12.5 Hz.
0111	13 Hz.	11.11 Hz.
0110	12 Hz.	10 Hz.
0101	11 Hz.	9.09 Hz.
0100	10 Hz.	8.33 Hz.
0011	9 Hz.	7.69 Hz.
0010	8.5 Hz.	7.14 Hz.
0001	8 Hz.	6.66 Hz.
0000	7.5 Hz.	6.25 Hz.

f. Refresh External Register (RFEXT) - SCB:177735



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-4	not used	
3	EXTFLG	This bit is used to read a condition external to the Picture System. This can be used to read the status of special user designed external hardware. Reads zero when no device is attached. Read only.
2	EXTOUT	This bit is available to control a device external to the Picture system. This can be used to send information to special user designed hardware. Cleared by RESET.
1	NORTI	This bit is set to cause the Refresh Processor to disable the Remote Terminal Interface (RTI). In this mode the RTI will not send data to the Line Generator. Normally used for diagnostic purposes. Cleared by RESET.
0	EXTSEL	When this bit is clear the Refresh Processor synchronization signals come from a line clock generated by the Picture System A.C. power source. When this bit is set, the synchronization signals come from another source external to the Picture System. This allows

customer designed hardware to
supply the Refresh Processor
synchronization signals.

2.4.2.2.1 Refresh and Device Processor Data Structures

The Refresh and Device Processor (Refresh Processor) is a completely independent microprocessor that controls the refreshing (or display) of data, the polling of Picture System devices and other special functions. The Picture Memory is used to contain data structures which the Refresh Processor interprets to determine the functions it is to perform. These data structures are the Refresh Table, Command & Parameter Tables and Display Lists/Control Tables as shown in Figure 2.4-4.

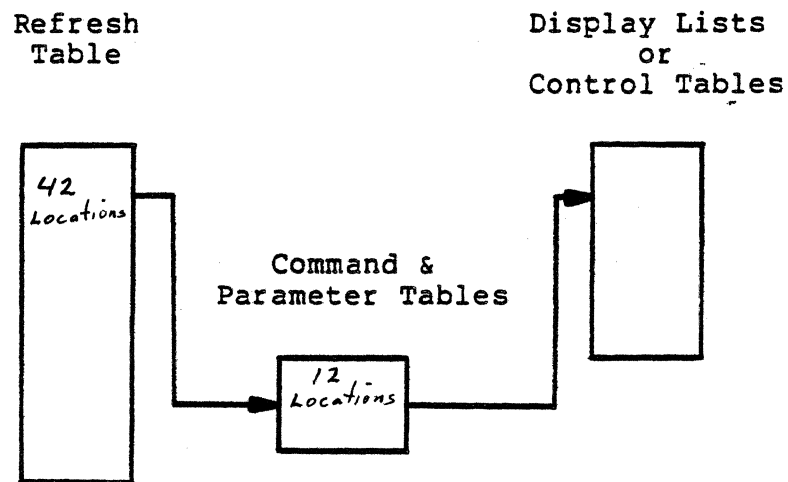


Figure 2.4-4
Refresh and Device Processor Data Structures

Upon RESET, the Refresh Processor is disabled while the Picture Memory resident data structures are loaded under software control from the Picture Controller. When the data structures have been loaded into the Picture Memory, the Refresh Processor is enabled by loading the address of the Refresh Table into the RFDT register (see Section 2.4.2.2d) and acknowledging the Refresh Processor Request (RP REQ) bit in the System Interrupt Request Register (see Section 2.5.2). The Refresh Processor will then begin to scan the entries of the Refresh Table, interpreting the entries either as a command (see below) or as the address of a Command & Parameter Table which specifies the function to perform. As the Refresh Processor executes the commands of the Refresh Table, the RP REQ bit is set in the System Interrupt Request Register as a command causes an interrupt to be requested. When the Refresh Table has been completely scanned, the Refresh Processor then waits for the next line frequency synchronization signal to begin scanning the Refresh Table again. The data structures utilized by the Refresh and Device Processor are detailed below.

- a. Refresh Table (RFTAB): [address specified by software]

The Refresh Table is a block of 42 contiguous locations in Picture Memory. It is organized as shown in Figure 2.4-5.

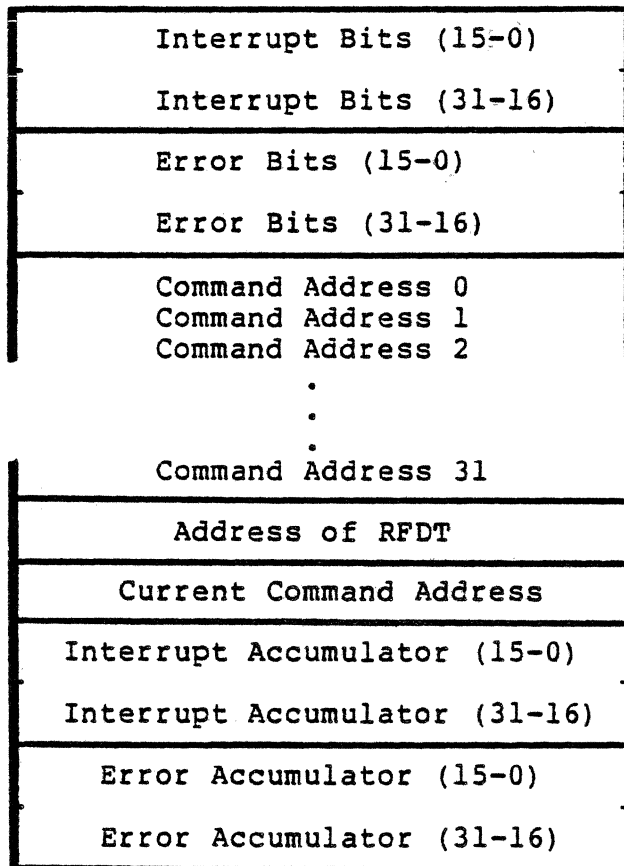


Figure 2.4-5
Refresh Table Organization

The first four words in the Refresh Table contain 32 Interrupt Bits and 32 Error Bits. These bits correspond with the 32 Command Addresses (Interrupt Bit 0 and Error Bit 0 with Command Address 0, etc.). When a Command Address points to a command that causes an interrupt condition, the corresponding Interrupt Accumulator bit is set. If a detectable error occurs, both the corresponding Interrupt Accumulator bit and Error Accumulator bit are set. If the RP REQ bit is not set in the System Interrupt Request Register, then the Inter-

rupt Accumulator 31-0 and Error Accumulator 31-0 are transferred to the Interrupt Bits 31-0 and Error Bits 31-0 and the RP REQ bit is set to generate an interrupt request to the host Picture Controller. If the RP REQ bit is set in the System Interrupt Request Register, then the Refresh Processor continues to the next Command Address until the RP REQ bit is sensed to have been acknowledged from the previous interrupt. At that time, the Interrupt Accumulator 31-0 and Error Accumulator 31-0 are transferred to the Interrupt Bits 31-0 and Error Bits 31-0 and the RP REQ bit is set to generate an interrupt request to the host Picture Controller. In both cases, after the Interrupt Accumulator 31-0 and Error Accumulator 31-0 are transferred to the Interrupt Bits 31-0 and Error Bits 31-0, the accumulator bits are cleared.

The next 32 locations of the Refresh Table are Command Addresses. The Refresh Processor scans these locations interpreting each as an address in Picture Memory where a Command & Parameter Table resides or as a special command. The Refresh Processor then performs the function specified. When the Refresh Processor completes each Command Address operation, it continues with the next Command Address. After each complete cycle, (all Command Addresses scanned), the Refresh Processor then waits for the synchronization signals that are controlled by the RFCLK and RFEXT registers before beginning to scan the Refresh Table again.

In addition to locations in Picture Memory, the Command Addresses may also contain the values 177777, 177776, 177775 and 177774. Because these addresses are in the SCB, Command & Parameter Tables cannot be located there. This allows these values to be used for special commands. A Command Address of 177777 is interpreted by the Refresh Processor as the "last" Command Address in the Refresh Table. This allows the Refresh Processor to scan less than all 32 Command Addresses. For example, if the twentieth Command Address is equal to 177777, then the Refresh Processor will scan only the first 19 Command Addresses. A Command Address of 177776 is interpreted by the Refresh Processor as a No Operation command (NOP). When the Refresh Processor finds this command it immediately continues to the next Command Address. Normally the Refresh Table is initialized to contain 32 NOPs (177776). When a Command & Parameter Table is to be added, it is written into Picture Memory and a NOP is changed to point to the new Command & Parameter Table. Similarly, when a Command & Parameter Table is deleted, it's entry in the Refresh

Table is replaced with a NOP. A Command Address of 177775 sets the corresponding Interrupt and Error Bits in the Refresh Table and then sets the RP REQ bit in the System Interrupt Request Register. The Refresh Processor then waits until the RP REQ bit is acknowledged before continuing to the next command in the Refresh Table. This command is normally used for diagnostic purposes. A Command Address of 177774 causes the corresponding Interrupt and Error Bits in the Refresh Table and the RP REQ bit to be set in the System Interrupt Request Register. The Refresh Processor then proceeds to the next command in the Refresh Table without waiting for the RP REQ bit to be acknowledged. This command can be used to synchronize to the A.C. power source.

The location in the Refresh Table after the 32nd Task Pointer is used to hold the address of the Refresh Processor Data Register (RFDT). The Refresh Processor uses this information when it is writing to the Line Generator Input Controller. This allows identical Refresh Processor hardware to be used in multiple Line Generator applications.

The next five locations in the Refresh Table are used by the Refresh Processor hardware and are not normally written by the user. The first of these locations is the Current Command Address. The Refresh Processor uses this location to store the address in Picture Memory of the Command Address whose Command & Parameter Table is currently being executed. The value in this location can be used to diagnose which Command Address was currently being executed should the Refresh Processor fail to complete a command. For example, if the Refresh Table is located at 1000(octal) and the Command & Parameter Table of Command Address 5 is currently being executed, the value in this location will be $1000 \text{ (Refresh Table base address)} + 4 \text{ (for the Interrupt and Error Bits)} + 5 \text{ (Command Address 5)} = 1000 + 4 + 5 = 1011 \text{ (octal)}$. The last four locations in the Refresh Table are the Interrupt Accumulator and the Error Accumulator. These locations are used to buffer any interrupts that occur while the user is servicing previously initiated interrupts. When a command causes an interrupt to be requested, the Refresh Processor will first set the corresponding bits in these 4 locations. If the RP REQ bit is set in the System Interrupt Request Register, the Refresh Processor continues to the next Command Address. If it is clear, the Interrupt and Error Accumulators are transferred to the Interrupt Bit and Error Bit locations in the Refresh Table and the RP

REQ bit is set in the System Interrupt Request Register. After the accumulators are transferred they are zeroed to allow the next set of several interrupts to accumulate while the last requests are processed by the Picture Controller's System interrupt service routine.

b. Command & Parameter Table

A Command & Parameter Table is a 12-word block of Picture Memory that contains 11 parameter words and a Command Word. The parameters required vary dependent upon the specified Command Word. As shown in Figure 2.4-6, the Command Address in the Refresh Table always points to (i.e., contains the address of) the 12th word of the Command & Parameter Table. The normal format of the Command Word is shown in Figure 2.4-7.

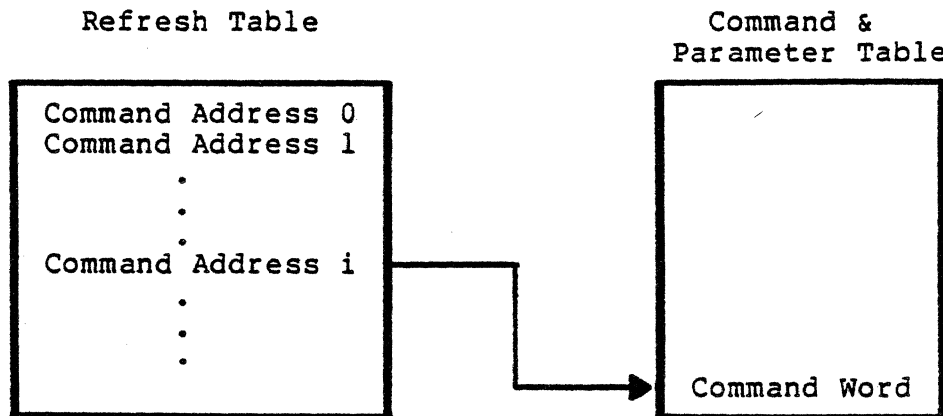


Figure 2.4-6
Command Word of the Command & Parameter Table

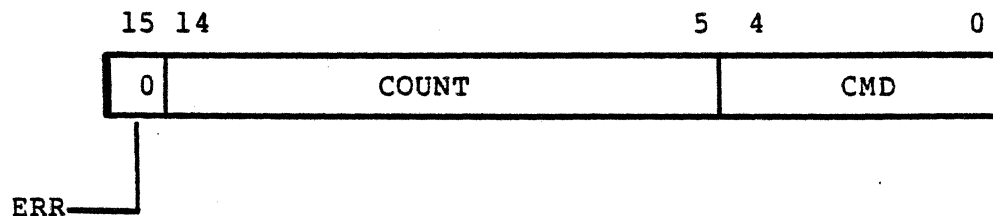


Figure 2.4-7
Format of the Normal Command Word

As Figure 2.4-7 shows, the normal Command Word contains three fields; CMD (bits 4-0), COUNT (bits 14-8) and ERR (bit 15, normally zero). The 5-bit CMD field is used to specify which display or device control function the Refresh Processor is to perform. The valid values for the CMD field and the associated Function Names are shown in Table 2.4-1. The functions performed by the Refresh Processor for each of these commands are detailed in Section 2.4.2.2.2.

Table 2.4-1
Command Word CMD Values

<u>CMD</u>	<u>Function Name</u>
0	NOP
1*	REFRESH
2	INITIALIZATION
3	ADD
4	REPLACE
5	COMPACT
6*	COMPACT2
7	DELETE/COMPACT
10*	DELETE2
11	REPLACE/COMPACT
12*	REPLACE2
13	CHANGE STATUS
14	CURSOR
15	LIGHT PEN
16	DEVICE CONTROL (interrupt initialization)
17	DEVICE CONTROL (with interrupts)
20	DEVICE CONTROL (without interrupts)
21-37	Not Assigned

*Normally these functions are initiated only by the Refresh Processor.

PS2/MPS Reference Manual
Hardware Details

The 10-bit COUNT field is an unsigned integer that allows a command to execute a fixed number of times and interrupt upon completion. The following list describes how each command uses the COUNT field.

<u>CMD</u>	<u>COUNT</u>	<u>FUNCTION</u>
0 NOP	=0	Execute a normal NOP.
	>0	Execute a NOP and decrement COUNT until it is equal to zero then set COUNT to zero and set the corresponding Interrupt Accumulator bit in the Refresh Table.
1 REFRESH	=0	Execute a normal REFRESH.
	>0	Refresh the display list and decrement COUNT every cycle until COUNT is equal to zero, then change to a NOP command with a zero COUNT field and set the corresponding Interrupt Accumulator bit in the Refresh Table.
2 INITIALIZATION	=0	Execute a normal INITIALIZA-TION.
	>0	Execute a normal INITIALIZA-TION but set the COUNT field in the REFRESH command that is generated to COUNT-1. Then continue normally.
3 ADD	=0	Execute a normal ADD.
	>0	Execute a normal ADD but set the COUNT field in the REFRESH command that is generated to COUNT-1. Then continue nor-mally.

PS2/MPS Reference Manual
Hardware Details

<u>CMD</u>	<u>COUNT</u>	<u>FUNCTION</u>
4 REPLACE	=0	Execute a normal REPLACE.
	>0	Execute a normal REPLACE but set the COUNT field in the REFRESH command that is generated to COUNT-1. Then continue normally.
5 COMPACT	=0	Display while compacting and set the corresponding Interrupt Accumulator bit in the Refresh Table upon completion.
	>0	Do not display while compacting. After the COMPACT command finishes, write out a REFRESH command with a COUNT field equal to COUNT and continue normally.
6 COMPACT2	=0,>0	This command is initiated only by the hardware and does not use the COUNT field.
7 DELETE/COMPACT	=0	Display while deleting and compacting and set the corresponding Interrupt Accumulator bit in the Refresh Table upon completion.
	>0	Do not display while executing the DELETE/COMPACT command then write out a REFRESH command with a COUNT field equal to COUNT and continue normally.
10 DELETE2	=0,>0	This command is initiated only by the hardware and does not use the COUNT field.

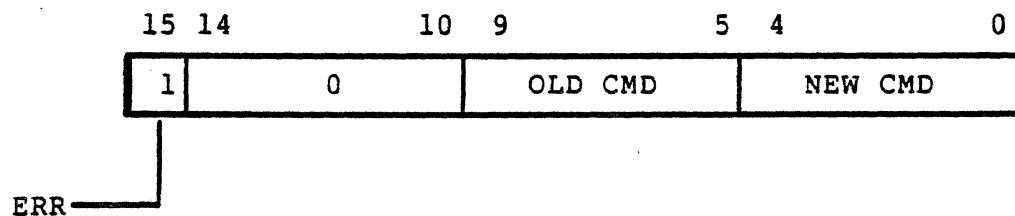
PS2/MPS Reference Manual
Hardware Details

<u>CMD</u>	<u>COUNT</u>	<u>FUNCTION</u>
11 REPLACE/COMPACT	=0	Display while replacing and compacting and set the corresponding Interrupt Accumulator bit in the Refresh Table upon completion.
	>0	Do not display while executing the REPLACE/COMPACT command. Then write out a REFRESH command with a COUNT field equal to COUNT and continue normally.
12 REPLACE2	=0,>0	This command is initiated only by the hardware and does not use the COUNT field.
13 CHANGE STATUS	=0	Display while changing the status and set the corresponding Interrupt Accumulator bit in the Refresh Table upon completion.
	>0	Do not display while the CHANGE STATUS executes. Then write out the REFRESH command with a COUNT field equal to COUNT and continue normally.
14 CURSOR	=0	Execute the CURSOR command normally.
	>0	Execute the CURSOR command and decrement COUNT every cycle until COUNT is equal to zero. Then change to a NOP command with a zero COUNT field and set the corresponding Interrupt Accumulator bit in the Refresh Table.

PS2/MPS Reference Manual
Hardware Details

<u>CMD</u>	<u>COUNT</u>	<u>FUNCTION</u>
15 LIGHT PEN	=0	Execute the LIGHT PEN command normally.
	>0	Execute the LIGHT PEN command and decrement COUNT every cycle until COUNT is equal to zero. Then change to a NOP command with a zero COUNT field and set the corresponding Interrupt Accumulator bit in the Refresh Table.
16,17,20 DEVICE CONTROL	=0	Execute the DEVICE CONTROL command normally.
	>0	Execute the DEVICE CONTROL command and decrement COUNT every cycle until COUNT is equal to zero, then change to a NOP command with a zero COUNT field and set the corresponding Interrupt Accumulator bit in the Refresh Table.
21-37	0	Not assigned.

If during the execution of a command the Refresh Processor detects an error (bad parameters, no Picture Memory, etc.) it will set both an Error Bit and an Interrupt Bit in the Refresh Table and set the RP REQ bit in the System Interrupt Request Register. In addition, because some errors might leave the display list in an uncertain state, the Refresh Processor will change the CMD word in the current Command & Parameter Table to allow other Refresh Processor activities to continue normally. The resulting Error Command Word is detailed below.



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	ERR	This bit is set when the Refresh Processor has detected an error condition.
14-10	Not used	Always zero.
9-5	OLD CMD	These five bits are the CMD bits of the command that caused the error.
4-0	NEW CMD	These five bits are generated by the Refresh Processor so that other Refresh Processor activities can continue normally. Table 2.4-2 lists the error conditions associated with each valid command, along with the corresponding NEW CMD fields.

Table 2.4-2
Command Word CMD Field After An Error

<u>CMD</u>	<u>NAME</u>	<u>ERROR CONDITION</u>	<u>NEW CMD</u>
0	NOP	No error conditions	-
1	REFRESH	No error conditions	-
2	INITIALIZE	Not enough memory	NOP
3	ADD	No error conditions	-
4	REPLACE	No error conditions	-
5	COMPACT	Name not found	REFRESH
6	COMPACT2	Name not found 2nd pass	NOP
7	DELETE/COMPACT	Name not found	REFRESH
10	DELETE2	Name not found 2nd pass	NOP
11	REPLACE/COMPACT	Name not found	REFRESH
12	REPLACE2	Name not found 2nd pass	NOP
13	CHANGE STATUS	Name not found	REFRESH
14	CURSOR	No error conditions	-
15	LIGHT PEN	No error conditions	-
16-20	DEVICE CONTROL	No error conditions	-
21-37	Not assigned	Illegal Command	NOP

c. Display Lists and Control Tables

Each Command & Parameter Table may have associated display lists or control tables. These data structures reside in Picture Memory and must be initialized either by the Picture Controller software or the Refresh Processor. The display lists and control tables associated with each of the valid commands are detailed in the following section.

2.4.2.2.2 Refresh and Device Processor Commands

The Refresh and Device Processor interprets the data structures described in Section 2.4.2.2.1 and performs the functions designated by the Command Word of the Command & Parameter Tables. The valid commands that may be specified are:

- a. NOP
- b. REFRESH
- c. INITIALIZATION
- d. ADD
- e. REPLACE
- f. COMPACT
- g. DELETE/COMPACT
- h. REPLACE/COMPACT
- i. CHANGE STATUS
- j. CURSOR
- k. LIGHT PEN
- l. DEVICE CONTROL

This section details each of these commands. Throughout this section, several conventions are used in describing the operation of the commands. Each command will show a diagram of the Command & Parameter Table in which the command resides. The contents of the Command & Parameter Table will be shown both "Before" and "After" execution of the command. The status of the parameters will be indicated by the following mnemonics:

Mnemonic Definition

NU	Not Used.
NC	Used but Not Changed.
T	Temporary (data may change).
SH	Parameter (written by Software, used by Hardware).
HS	Parameter (written by Hardware, read by Software).
P	Protected (software should not change data).
(A)	Comment applies to parameter after execution.
(B)	Comment applies to parameter before execution.

If neither an (A) nor a (B) comment appear, the parameter has the same meaning both before and after the execution of the command.

The HALT and JUMP commands referred to below are Multi Picture System Refresh Control Commands. See Section 2.4.3c for further details of these commands.

PS2/MPS Reference Manual
Hardware Details

- a. NOP COMMAND (0)
No Operation.

	Before	After
CMD	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NU	NU
	NOP	NOP

The NOP command performs no operation. No parameters are used and no interrupt is generated.

PS2/MPS Reference Manual
Hardware Details

- b. REFRESH COMMAND (1)
Used to refresh the display list.

	Before	After	
	T	T	
	T	T	
MINLIM	P	P	Bottom of Display Memory
MAXLIM	P	P	Top of Display Memory
RP	P	P	Address to start Display
	NU	NU	
AVSGBG	HS	HS	Bottom of Available Memory
AVSGND	HS	HS	Top of available Memory
BEGSEG	HS	HS	Address to start Adding new data
	NU	NU	
	NU	NU	
CMD	REFRESH	REFRESH	

The Refresh Command starts the Refresh Processor at RP, where it begins refreshing by reading commands from Picture Memory and sending them to the Line Generator. This reading continues sequentially through memory until the Refresh Processor finds a JUMP command. This causes it to continue refreshing at the place specified by word 2 of the JUMP. If the Refresh Processor reads all the data up to MAXLIM, it wraps around to MINLIM and continues to read. The Refresh Processor refreshes until it finds a HALT command. When a HALT command is encountered, the Refresh Processor continues with the next Command Address in the Refresh Table.

- c. INITIALIZATION COMMAND (2)
Used to initialize the display list.

Before		
	NU	
	NU	
MINLIM	SH	Bottom of Display Memory
MAXLIM	SH	Top of Display Memory
	T	
LG1	SH	(B) Line Generator set up word 1
LG2	SH	(B) Line Generator set up word 2
LG3	SH	(B) Line Generator set up word 3
LG4	SH	(B) Line Generator set up word 4
LG5	SH	(B) Line Generator set up word 5
LG6	SH	(B) Line Generator set up word 6
CMD	INIT	

After		
	NU	
	NU	
MINLIM	P	Bottom of Display Memory
MAXLIM	P	Top of Display Memory
RP	P	(A) Address to Start Display
	NU	
AVSGBG	HS	(A) Bottom of Available Memory
AVSGND	HS	(A) Top of Available Memory
BEGSEG	HS	(A) Address to Add next segment
ENDSEG	HS	(A) Address to Add next segment
	NU	
CMD	REFRESH	

The INITIALIZATION command takes MINLIM and MAXLIM as input parameters and allocates the Picture Memory between them for use as a display list. It also uses the LG1-LG6 parameters to build the rudimentary display list shown in Figure 2.4-9. Since the Picture Generator requires 32 bits of data per command (2 16-bit words), the number of words between MINLIM and MAXLIM must be even. The Refresh Processor subtracts MINLIM from MAXLIM and if the result is odd, MAXLIM is decremented so there will be an even number of words in the display list.

If the check of MINLIM and MAXLIM shows that there is not sufficient memory between them for a display list (10 words), then the ERR bit is set in the CMD word, the CMD is changed to NOP and the error interrupt sequence is initiated. If no error occurs, then the CMD is changed to REFRESH and the done interrupt sequence is initiated. Given the Command & Parameter Table in Figure 2.4-8, the Refresh Processor will build the display list in Figure 2.4-9.

Before		After	
X	X	X	X
X	X	X	X
MINLIM	1500	MINLIM	1500
MAXLIM	1611	MAXLIM	1610
X	X	RP	1502
LG1	200	X	X
LG2	176000	AVSGBG	1512
LG3	134000	AVSGND	1606
LG4	173777	BEGSEG	1512
LG5	2	ENDSEG	1512
LG6	0	X	X
INIT	2	REFRESH	1

Figure 2.4-8
Example of a Command & Parameter Table
Before and After Execution of INITIALIZATION Command

The display list that would result from the execution of the Command & Parameter Table of 2.4-8 is shown below.

<u>Pointers</u>	<u>Symbolic Contents</u>	<u>Address</u>	<u>Contents</u>
MINLIM	HALT	1500/	40202
	X		X
RP	LG1		200
	LG2		176000
	LG3		134000
	LG4		173777
	LG5		2
	LG6		0
	JUMP		40203
	(MINLIM)	1511/	1500
AVSGBG/BEGSEG		1512/	X
			X
			X
			X
			X
AVSGND		1606/	X
			X
MAXLIM		1610/	

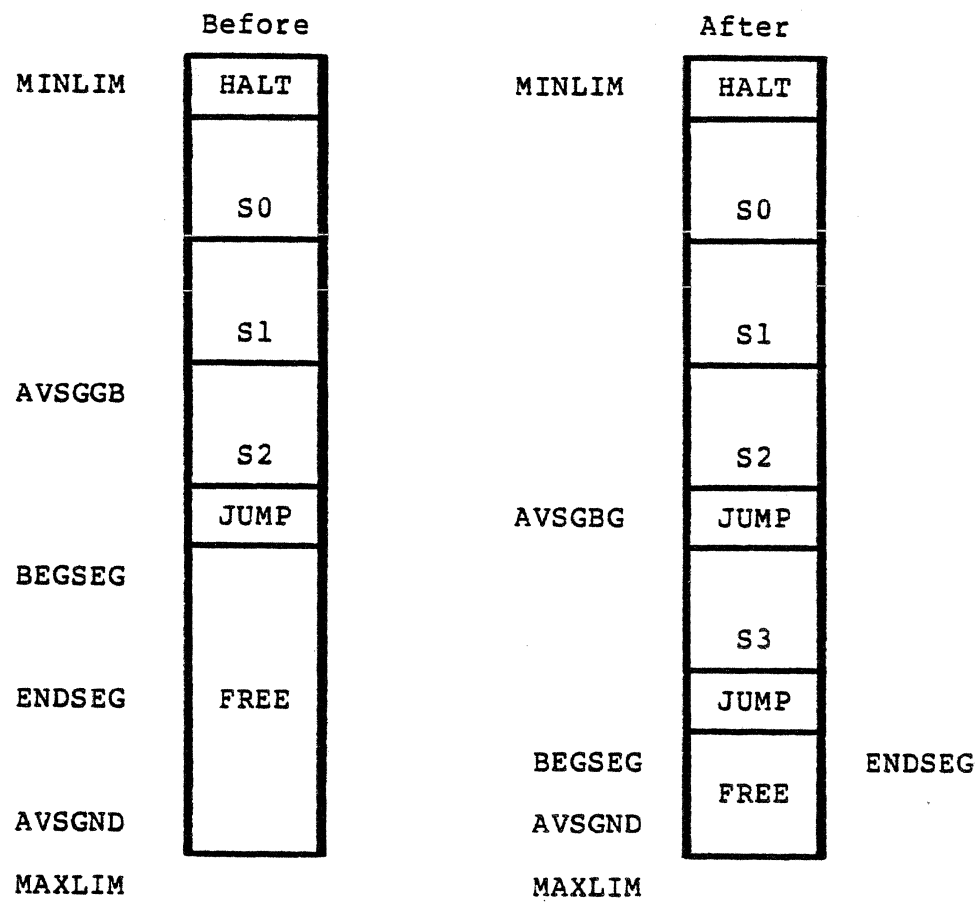
Figure 2.4-9
Initial Display List Generated By INITIALIZATION Command

- d. ADD COMMAND (3)
Used to add a new segment.

	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of Display Memory
MAXLIM	P	P	Top of Display Memory
RP	P	P	Address to start Display
	NU	NU	
AVSGBG	P	HS	(A) Beginning of New segment
AVSGND	P	HS	(A) End of Available Memory
BEGSEG	P	HS	(A) START address for next segment
ENDSEG	SH	NU	(B) End address for segment to be added. (A) Start address for next segment.
	NU	NU	
CMD	ADD	REFRESH	

The ADD command causes the Refresh Processor to put the new segment which lies between BEGSEG and ENDSEG into the display list. When the ADD is encountered, the CMD word is changed to REFRESH and the done interrupt sequence is initiated. The data in the display list (including the new segment) is then refreshed.

To use this command software reads the BEGSEG value from the Command & Parameter Table and writes the new segment into Picture Memory, starting at that value. The address of the first free memory location after the end of the new segment is written into ENDSEG and the CMD word is changed from REFRESH to ADD. When the Refresh Processor encounters the ADD command, it adds two to the contents of BEGSEG and moves it into AVSGBG, adds two to the contents of ENDSEG and places the result into BEGSEG and into ENDSEG. The JUMP command at the end of the previous segment is changed to JUMP to the new segment and a JUMP command is added at the end of the new segment to JUMP to AVSGND + 2. This allows the display of the new segment on subsequent refresh cycles.



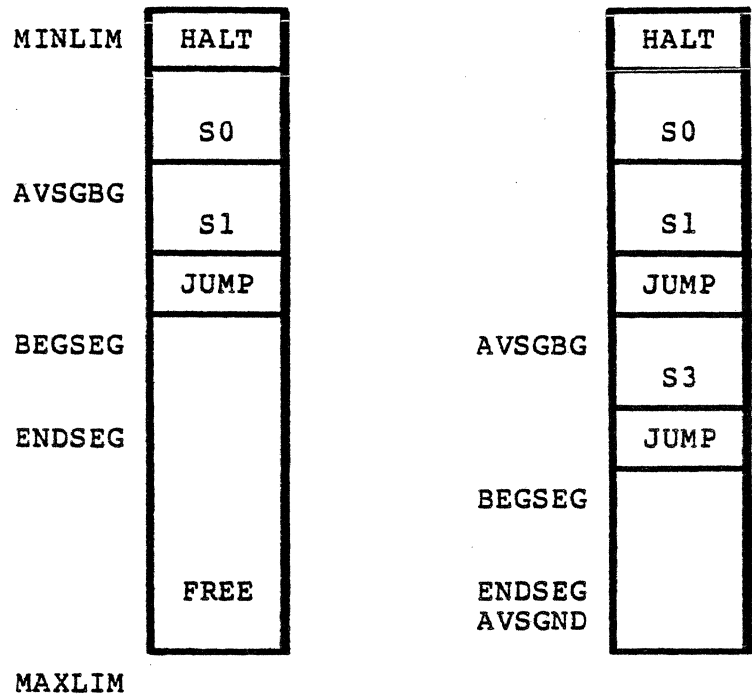
Display list before and after Add of S3

- e. REPLACE COMMAND (4)
Replaces an old segment with a new segment. Used in double buffering.

	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of Display Memory
MAXLIM	P	P	Top of Display Memory
RP	P	P	Address to Start Display
	NU	NU	
AVSGBG	P	HS	(B) Start of Segment to be replaced
AVSGND	P	HS	(A) End of Available Memory
BEGSEG	SH	HS	(B) Beginning address for new segment
ENDSEG	SH	HS	(B) End address for new segment.
			(A) Start address for next segment.
	NU	NU	
CMD	REPLACE	REFRESH	

The REPLACE command substitutes the new segment which lies between BEGSEG and ENDSEG for the segment that starts at AVSGBG. When the REPLACE is encountered, the Refresh Processor immediately changes the CMD word to REFRESH and initiates the done interrupt sequence. The data in the display list (including the new segment) is then refreshed. To implement double buffering using this command, software would first ADD the segment to be double buffered then read AVSGBG and AVSGND. These parameters are used as the bottom and top of the buffer area and the buffer area is divided into two parts. The first part begins at AVSGBG and contains the segment just added. The second part begins at the location half way between ABSGBG and AVSGND. Software writes the address of the beginning of the second part of the buffer into BEGSEG and starting at that address, writes the updated segment into Picture Memory. The address of the first free memory location after the end of the new segment is written into ENDSEG and the CMD word is changed from REFRESH to REPLACE. When the Refresh Processor encounters the REPLACE command, it puts two JUMP commands into Picture Memory. On subsequent refresh cycles the first JUMP command will cause the Refresh Processor to jump over the old segment and display the new segment and the second JUMP command will jump to the segment that follows AVSGND. The Refresh Processor adds two to ENDSEG and places the result into BEGSEG and into ENDSEG. The done interrupt sequence is then initiated. Doing the next update requires similar activity except that software moves the contents of AVSGBG into BEGSEG and places

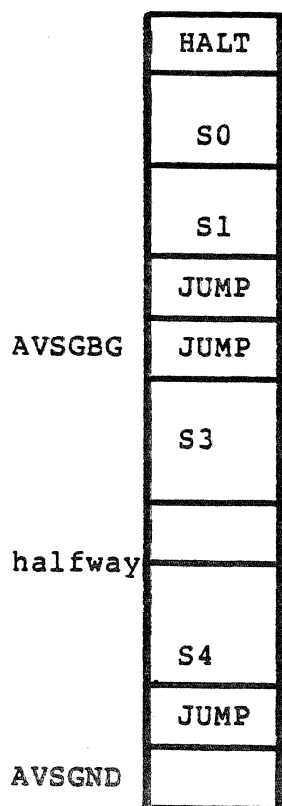
the updated segment into Picture Memory starting at AVSGBG. The following diagram illustrates the use of REPLACE for double buffering.



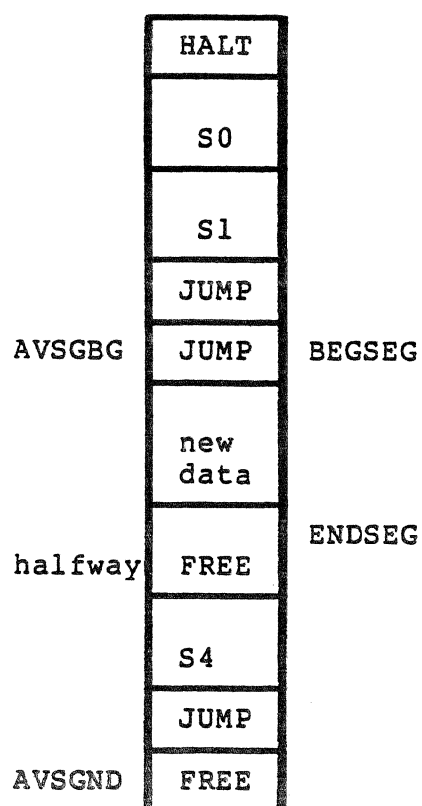
Display list before
ADDing the first
buffered segment.

Display list after ADDing
the first buffered segment
(S3).

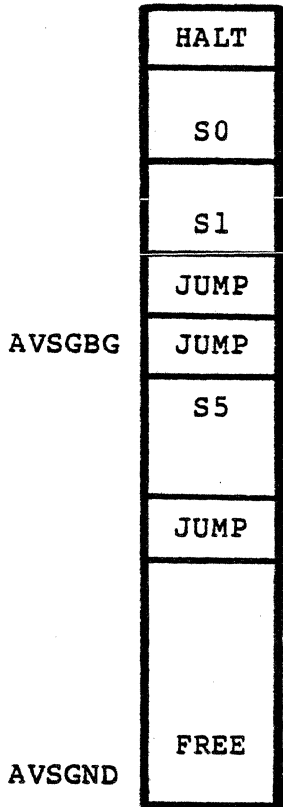
PS2/MPS Reference Manual
Hardware Details



Display list after
REPLACING S3 with
S4 (first buffer
swap)



Display list before
the second buffer
swap



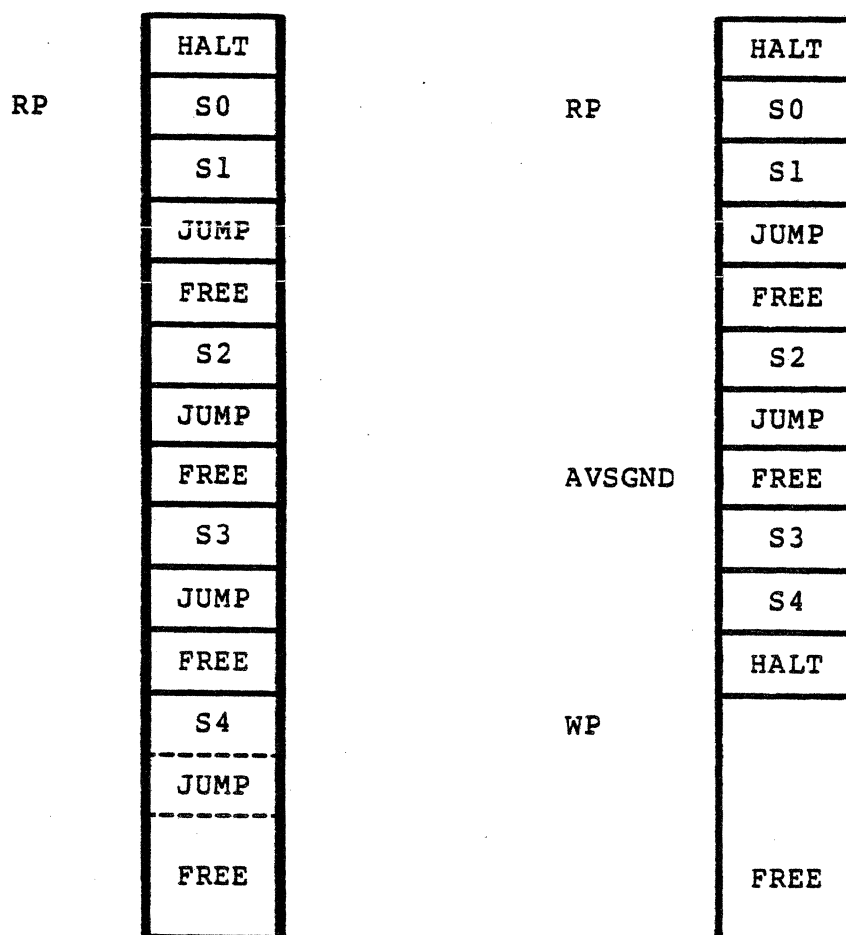
Display list after
second buffer swap

- f. COMPACT COMMAND (5)
Used to compact the free space in the display list.

	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of display list
MAXLIM	P	P	Top of display list
RP	P	P	Address to start display
WP	P	NU	Write pointer
AVSGBG	P	HS	(A) Beginning of compacted segment
AVSGND	P	HS	(A) End of available memory
BEGSEG	T	HS	(A) First location of free memory
ENDSEG	T	HS	(A) Start address for next segment
SEGNAM	SH	NU	(B) Name of segment to be compacted
CMD	COMPACT	REFRESH	

The COMPACT command searches the display list for the end of the segment specified by SEGNAM. The free memory is then collected and moved into the space after the end of that segment.

This command requires two display cycles. On the first cycle the Refresh Processor searches for the end of the segment. If the next command in the display list is a JUMP, then the free memory already resides after the segment and the free memory is not compacted and remains after the segment. The Refresh Processor then searches for the first non-JUMP command after the segment. The address of the command is decreased by two and the resulting value is placed in AVSGND. This will be used as the last location of the free memory by cycle 2. Data after this point, and up to the HALT, are moved so that no JUMP commands remain. The following illustrates the first cycle of the COMPACT command for segment 2 (S2).

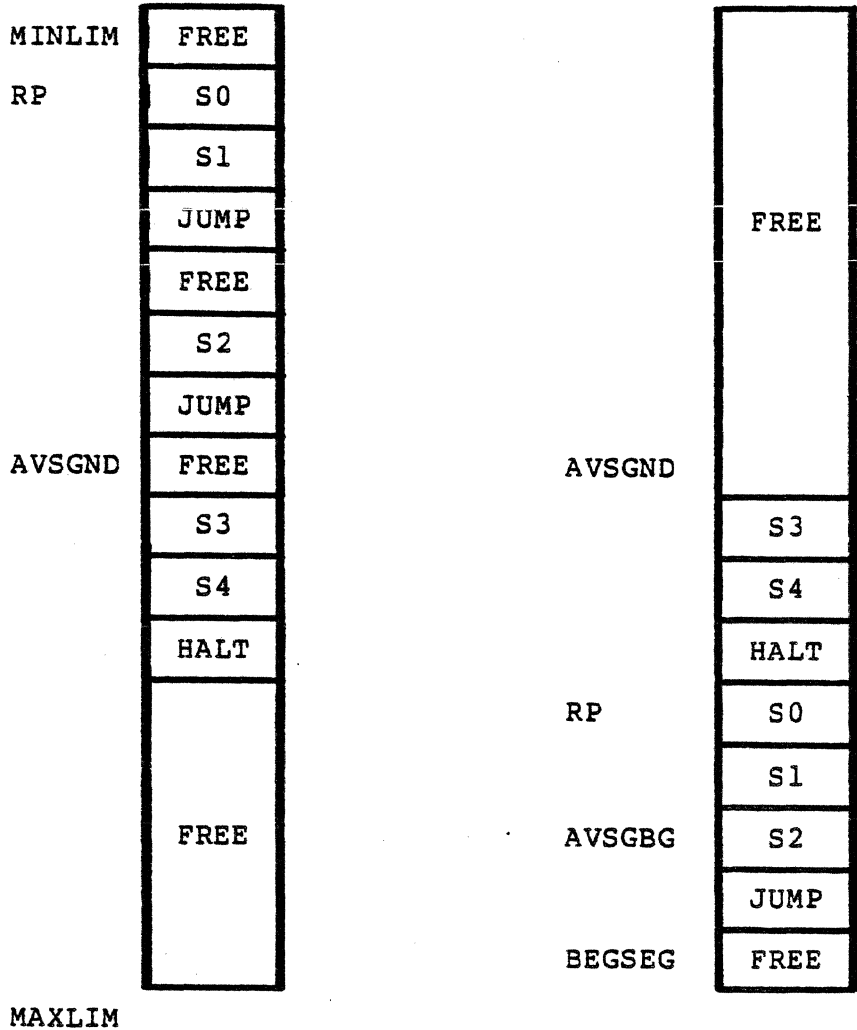


Display list before
COMPACT

Display list after
cycle 1 of COMPACT

COMPACT command with SEGNAM=S2

Cycle two of COMPACT is used to collect the remaining free memory. Refresh is started at RP and every word that is read is written back into locations starting at WP. (When either the read or the write reach MAXLIM, they wrap around to MINLIM). All JUMP commands are eliminated until the beginning of the segment to be compacted is found. The address of the location just after the beginning segment name is written into AVSGBG and the refresh continues to eliminate JUMPS until the end of the segment to be compacted. At this time, all the free space in the display list is located after the end of the segment and a JUMP to AVSGND+2 is inserted so that subsequent refresh cycles will jump around the free space. After the JUMP is inserted the address after the JUMP is written into BEGSEG and ENDSEG. The Refresh Processor changes the CMD to REFRESH and initiates a done interrupt sequence. The following illustrates the second cycle of the COMPACT command.



Display list after
cycle 1 of compact

Display list after
cycle 2 of compact

COMPACT command (cycle 2)

COMPACT checks for two errors. The first cycle is searching for a segment name (the contents of SEGNAM). If this name is not found the ERR bit is set in CMD, CMD is changed to REFRESH and the error interrupt sequence is initiated. On the second cycle if the segment name is not located, the ERR bit is set in the CMD word, the function in the CMD word is changed to NOP and the error interrupt sequence is then initiated.

- g. DELETE/COMPACT COMMAND (7)
Used to delete segments from the display list.

	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of display list
MAXLIM	P	P	Top of display list
RP	P	P	Address to start display
WP	P	NU	Write Pointer
AVSGBG	P	HS	(A) Beginning of available memory
AVSGND	P	HS	(A) End of available memory
BEGSEG	T	HS	(A) First location of free memory
ENDSEG	T	HS	(A) Start address for next segment
SEGNAM	SH	NU	(B) Name of segment to be deleted
CMD	DELETE/ COMPACT	REFRESH	

The DELETE/COMPACT command searches the display list for the segment specified by SEGNAM. This segment is then deleted and all the free memory in the display list is packed together and left in its place.

This command requires two display cycles. On the first cycle the Refresh Processor searches for the end of the segment. If the next command in the display list is a JUMP then there is already some free memory after the segment and it is left there. The Refresh Processor searches for the first non-JUMP command after the segment. The address of the command is decreased by two and the resulting value is placed in AVSGND. This will be used as the last location of the free memory by cycle 2. The transformed data after AVSGND and up to the HALT are moved so that no JUMP commands remain. The following illustrates the first cycle of the DELETE/COMPACT command for segment 2 (S2).

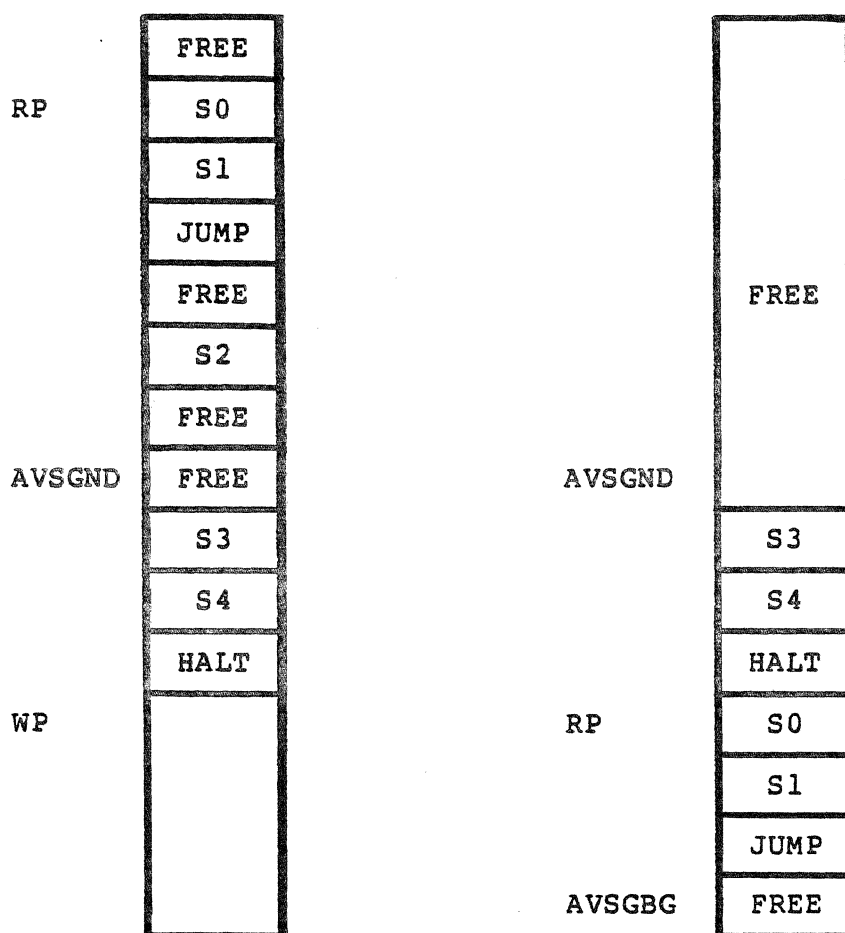
HALT		FREE
S0	RP	S0
S1		S1
JUMP		JUMP
FREE		FREE
S2		S2
JUMP		FREE
FREE	AVSGND	FREE
S3		S3
JUMP		S4
		HALT
S4	WP	
JUMP		

Display list before
cycle 1 of
DELETE/COMPACT

Display list after
cycle 1 of
DELETE/COMPACT

DELETE/COMPACT command with SEGNAM=S2

Cycle two of DELETE/COMPACT is used to collect the remaining free space. Refresh is started at RP and every word that is read is written back into the locations starting at WP. (When either the read or the write reach MAXLIM, they wrap around to MINLIM). JUMP commands are eliminated until the beginning of the segment to be deleted is found. At this point the segment name is replaced with a JUMP to AVSGND+2 (deleting the segment) and the address of the location after the JUMP command is written into AVSGBG, BEGSEG and ENDSEG. When the Refresh Processor finds the HALT command, it changes the CMD word to REFRESH and initiates a done interrupt sequence. The following illustrates the second cycle of the DELETE/COMPACT command.



Display list after
cycle 1 of DELETE/
COMPACT

Display list after
cycle 2 of DELETE/
COMPACT

DELETE/COMPACT Command (cycle 2)

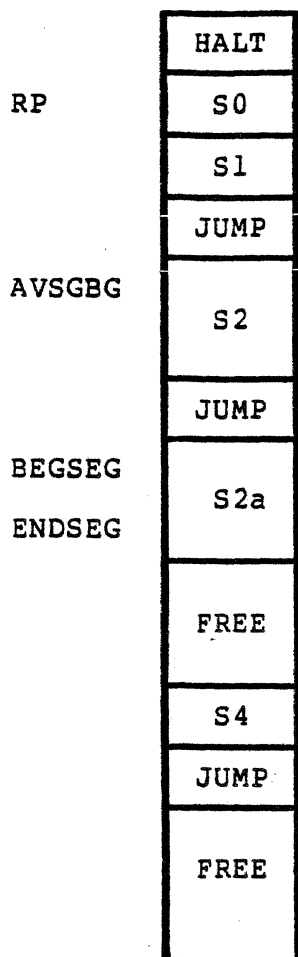
DELETE/COMPACT checks for two errors. The first cycle is searching for a segment name (the contents of SEGNAM). If this is not found, the ERR bit is set in CMD. CMD is then changed to REFRESH, and the error interrupt sequence is initiated. On the second cycle, the Refresh Processor again searches for the segment name. If it is not found, the ERR bit is set in CMD. CMD is then changed to NOP and the error interrupt sequence is initiated.

- h. REPLACE/COMPACT COMMAND (11)
Replaces on old segment with a new one and places all the free memory after the new segment.

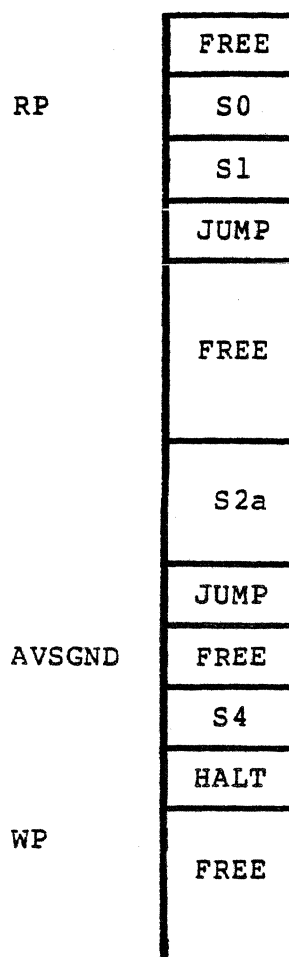
	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of display list
MAXLIM	P	P	Top of display list
RP	P	P	Address to start display
	NU	NU	
AVSGBG	P	HS	(B) Start of Segment to be replaced, (A) Beginning of new compacted segment.
AVSGND	P	HS	(A) End of Available Memory
BEGSEG	P	HS	(B) Beginning address for new segment, (A) First location of free memory.
ENDSEG	SH	HS	(B) End address for new segment, (A) Start address for next segment.
SEGNAM	SH	NU	(B) Name of the new segment
CMD	REPLACE COMPACT	REFRESH	

The REPLACE/COMPACT command replaces the segment that starts at AVSGBG with the segment that lies between BEGSEG and ENDSEG. After this replacement, all the free memory in the display list is packed together and left after the end of the new segment.

This command requires two display cycles. Before the first cycle starts the Refresh Processor writes a jump from the beginning of the old segment (pointed to by AVSGBG) to the beginning of the new segment (pointed to by BEGSEG) and a jump from the end of the new segment (pointed to by ENDSEG) to the beginning of the next (pointed to by AVSGND). This causes the new segment to replace the old segment. During the first cycle, the Refresh Processor searches for the end of the new segment. When this is found, refresh continues at the location after AVSGND. Data from this point up to the HALT command is moved so that no JUMP commands remain. The following illustrates the first cycle of REPLACE/COMPACT for segment 2 (S2).



Display list before
cycle 1 of
REPLACE/COMPACT

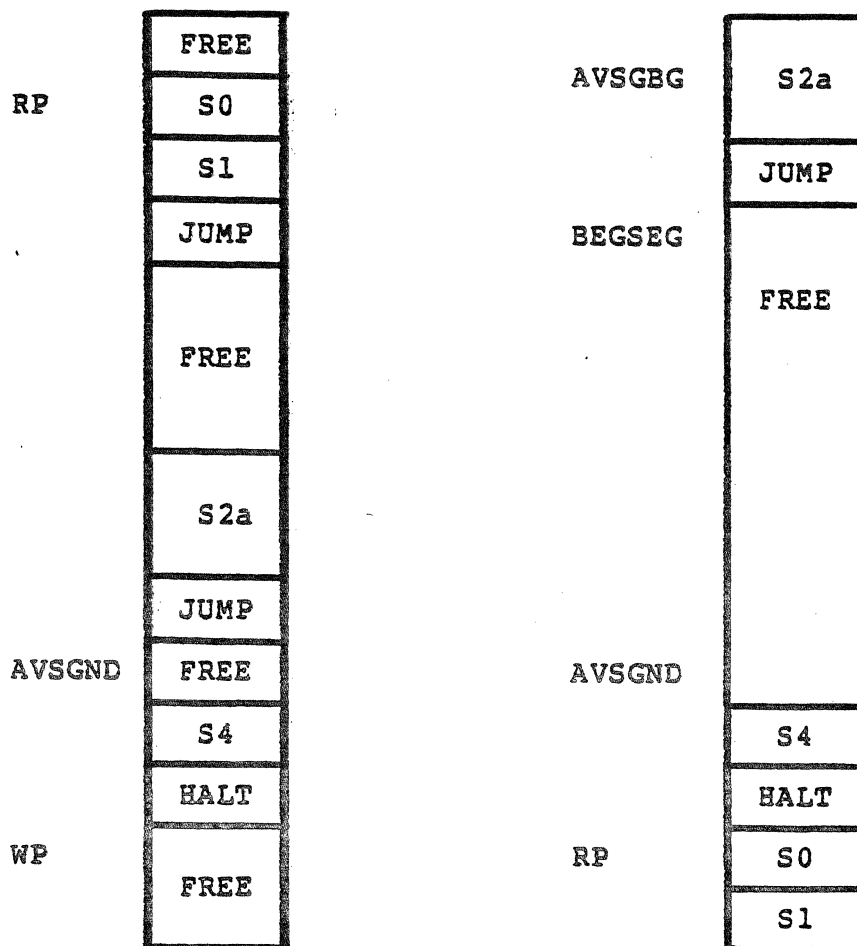


Display list after
cycle 1 of
REPLACE/COMPACT

REPLACE/COMPACT Command (Cycle 1)

Cycle two of REPLACE/COMPACT is used to collect the remaining free memory. Refresh is started at RP and every word that is read is written back into locations starting at WP. (When either the read or the write reach MAXLIM they wrap around to MINLIM.) All JUMP commands are eliminated until the beginning of the segment to be compacted is found. The address of the location just after the segment name is written into AVSGBG and the refresh continues to eliminate JUMPS until the end of the segment to be compacted is reached. At this time, all the free space in the display list is located at the end of the segment. A JUMP to AVSGND+2 is then inserted so that subsequent refresh cycles will jump around

the free space. After the JUMP is inserted, the address after the JUMP is written into BEGSEG and ENDSEG. The Refresh Processor changes the CMD to REFRESH and initiates a done interrupt sequence. The following illustrates the second cycle of the REPLACE/COMPACT command.



Display list after
cycle 1 of
REPLACE/COMPACT

Display list after
cycle 2 of
REPLACE/COMPACT

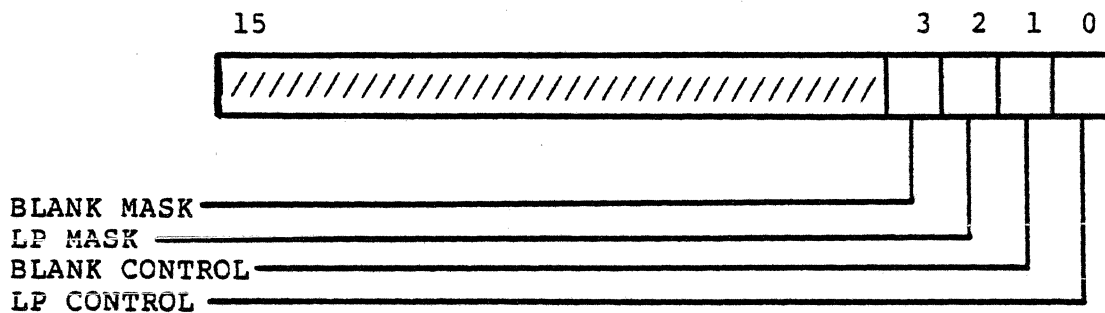
REPLACE/COMPACT Command (Cycle 2)

REPLACE/COMPACT checks for two errors. The first cycle is searching for a segment name (the contents of SEGNAM). If this name is not found, the ERR bit is set in CMD, CMD is changed to REFRESH and the error interrupt sequence is initiated. On the second cycle, if the segment name is not located, the ERR bit is set in the CMD word. The function in the CMD word is then changed to NOP and the error interrupt sequence is then initiated.

- i. CHANGE STATUS COMMAND (13)
Used to change the status of a segment (Light Pen Sensitivity or Segment Blank).

	Before	After	
	NU	NU	
	NU	NU	
MINLIM	P	P	Bottom of Display Memory
MAXLIM	P	P	Top of Display Memory
RP	P	P	Address to Start Display
	NU	NU	
AVSGBG	P	HS	
AVSGND	P	HS	
BEGSEG	P	HS	
PARAM	SH	NU	(B) Status change parameter
SEGNAM	SH	NU	(B) Segment to change
CMD	CHANGE STATUS	REFRESH	

This command searches for the segment name specified in SEGNAM and modifies it according to the contents of PARAM. The format for PARAM is detailed below.



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-4	not used	
3	BLANK MASK	This bit is set if the segment is to be BLANKED or UNBLANKED
2	LP MASK	This bit is set if the segment is to be Light Pen enabled or disabled.
1	BLANK CONTROL	If BLANK MASK is set, the segment specified in SEGNAME will be blanked when BLANK CONTROL is set and will not be blanked when BLANK CONTROL is zero. If BLANK MASK is zero, BLANK CONTROL has no effect.
0	LP CONTROL	If LP MASK is set, the segment specified in SEGNAME will be enabled as light pen sensitive when LP CONTROL is set and will not be enabled as light pen sensitive when LP CONTROL is zero. If LP MASK is zero LP CONTROL has no effect.

CHANGE STATUS checks for one error. If the name specified in SEGNAME is not found, the ERR bit in CMD is set. CMD is then changed to REFRESH and the error interrupt sequence is initiated.

- j. CURSOR COMMAND (14)
Used to display a cursor on the screen.

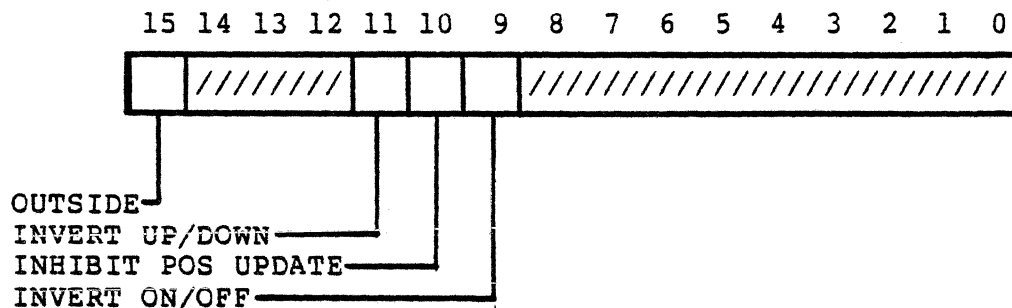
	Before	After	
LG1	SH	P	Line Generator STATUS
LG2	SH	P	used for scope select.
ON/OFF MASK	SH	P	These masks are used to
			determine which of four
UP/DOWN MASK	SH	P	cursors will be used.
MSKADR	SH	P	Picture System address
			for UP/DOWN and ON/OFF data.
CTLPTR	SH	P	Pointer to Cursor Control
			Table (multiple of 4)
XADR	SH	P	Picture System address for
			XPOS data.
YADR	SH	P	Picture System address for
			YPOS data.
XPOS	T	HS	X screen coordinate of
			Cursor center.
YPOS	T	HS	Y screen coordinate of
			Cursor center.
STATUS	SH&HS	SH&HS	Cursor control and status
			information.
CMD	CURSOR	CURSOR	

This command automatically displays a cursor at the screen coordinates (XPOS, YPOS). The Refresh Processor reads the address specified by MSKADR and this data is anded with ON/OFF MASK and if no bits remain on, no update of the cursor position will occur. If one or more bits remain on, the addresses specified by XADR and YADR are read and the contents of these locations are used in the following equations to generate XPOS and YPOS.

$$\begin{aligned} \text{XPOS} &= ((\text{contents of location XADR}) + \text{XOFFSET}) * (\text{XSCALE}) \\ \text{YPOS} &= ((\text{contents of location YADR}) + \text{YOFFSET}) * (\text{YSCALE}) \end{aligned}$$

XPOS is then checked against XMAX and XMIN. If it is greater than XMAX, its reduced to XMAX and if it is less than XMIN it is increased to XMIN. Similarly YPOS is checked and brought within YMAX and YMIN. If either XPOS or YPOS is modified by these range checks bit 15 in the STATUS word is set. The format of STATUS is detailed below.

PS2/MPS Reference Manual
Hardware Details



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	OUTSIDE	This bit is set by the Refresh Processor to indicate that either XPOS or YPOS was out of range.
14-12 not used		
11	INVERT UP/DOWN	This bit is set by software if the result of the UP/DOWN test is to be inverted.
10	INHIBIT POS UPDATE	This bit is set by software to inhibit the updating of the values of XPOS and YPOS. This mode allows the cursor to be moved by software. If the result of the ON/OFF test is not zero, this bit has no effect.
9	INVERT ON/OFF	This bit is set by software if the result of the ON/OFF test is to be inverted.
8-0 not used		

The parameters XOFFSET, XSCALE, XMAX, XMIN, YOFFSET, YSCALE, YMAX and YMIN are located in the Cursor Control Table (CTLPTR points to the Cursor Control Table). The following shows the contents of this table. The Cursor Control Table must be located in Picture Memory on a four word boundary (the two low-order address bits must equal zero).

CTLPTR

XOFFSET
XSCALE
XMAX
XMIN
YOFFSET
YSCALE
YMAX
YMIN
CURS0R0
CURS0R1
CURS0R2
CURS0R3

The entries CURS0R0 - CURS0R3 are pointers to four display lists which describe four cursors. The Refresh Processor generates a two-bit number and uses this number to select a cursor. The low-order bit of this number is computed by reading the Picture System Address pointed to by MSKADR and by anding the contents with UP/DOWN MASK. If the result of this operation is not zero, the low-order bit is set. The high-order bit is computed by reading the Picture System Address pointed to by MASKADR and by anding the contents with ON/OFF MASK. If the result of this test is not zero, the high-order bit is set. The low-order bit is inverted if bit 9 of STATUS is on and the high-order bit is inverted if bit 11 of STATUS is on. The resulting 2 bit number selects one of the four cursors (CURS0R0 for a value of 0, CURS0R1 for value of 1, etc.). When the cursor is selected the Refresh Processor sends LG1 and LG2 to the Line Generator to perform the scope select function. XPOS and YPOS are then used to move the beam to screen coordinate (XPOS, YPOS) and the Line Generator is placed in Relative Mode. The Refresh Processor then begins to display the cursor. Display continues until a Refresh Control HALT Command is encountered.

The following examples all use the Command & Parameter Table of Figure 2.4-10 to illustrate the control of the Cursor. The Cursor Control Table is assumed to have CURSOR0 defined as a dim cursor, CURSOR1 defined as a bright cursor and CURSOR2 and CURSOR3 defined as a null display.

LG1	200	
LG2	176000	Select all scopes
ON/OFF MASK	(see examples)	
UP/DOWN MASK	(see examples)	
MSKADR	177664	Standard Tablet Status Register
CTLPTR	177720	Cursor Control Table address
XADR	177665	Standard Tablet X address
YADR	177666	Standard Tablet Y address
XPOS	1000	Initial X screen position =1000(8)
YPOS	177700	Initial screen position =-100(8)
STATUS	(see examples)	
CMD	14	Cursor Command

Figure 2.4-10
Example Task Command & Parameter Table for Cursor Control

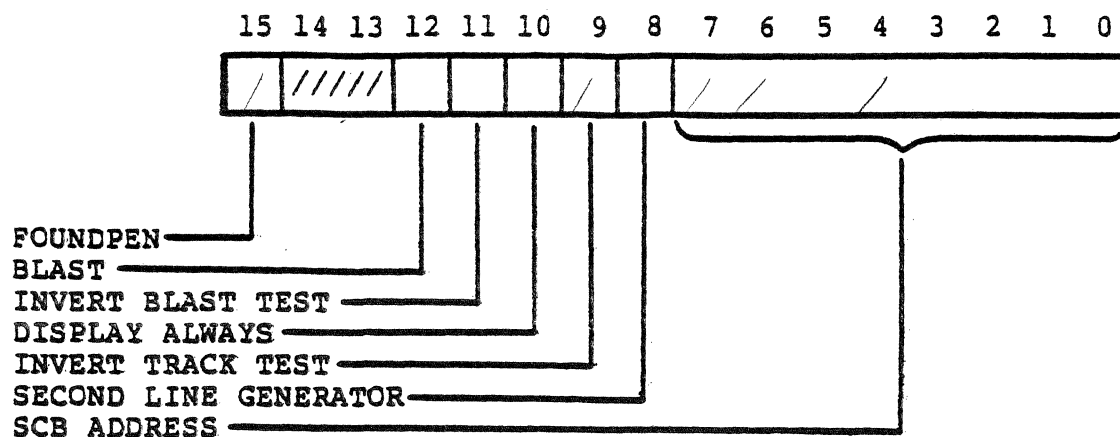
PS2/MPS Reference Manual
Hardware Details

<u>STATUS</u>	<u>ON/OFF</u>	<u>UP/DOWN</u>	<u>Function</u>
001000	100007	000020	Track the cursor according to the tablet coordinates, displaying no cursor when the pen is out of proximity, a dim cursor when the pen is in proximity but not down and a bright cursor when the pen is pressed down in proximity. The pen is in proximity when the Data Tablet Status Register does not have bit 15, 2, 1, or 0 set (i.e., 100007). The pen is down when the Data Tablet Status Register has bit 4 set (i.e., 000020).
007000	100000	000020	Inhibit cursor tracking, displaying no cursor when the pen is in proximity, a dim cursor when the pen is pressed down but not in proximity and a bright cursor when the pen is not in proximity and not pressed down.

- k. LIGHT PEN COMMAND (15)
Used to locate and track the Light Pen.

	Before	After	
LG1	SH	P	Line Generator Status
LG2	SH	P	Command (used for scope select)
TRACK MASK	SH	P	Mask to control tracking
BLAST MASK	SH	P	Mask to control screen blast
MSKADR	SH	P	Pointer to address for BLAST & TRACKING MASKS
HITMASK	SH	P	Light pen hit bit mask
HDX	SH	P	Zeroed by user of first call to Light Pen and untouched by user thereafter
HDY	SH	P	
XPOS	SH	HS	X screen coordinate of tracking cross center
YPOS	SH	HS	Y screen coordinate of tracking cross center
STATUS	SH&HS	SH&HS	Control and status information
CMD	LIGHTPEN	LIGHTPEN	

This command is used to control two forms of interaction with the Light Pen. The first form of interaction, tracking, allows software to locate the Light Pen by putting up a small figure (i.e., a tracking cross). If there are hits on the figure, the Refresh Processor moves the center of the figure to the position indicated by the hits. This position (XPOS,YPOS) is used on the next cycle as the new center of the figure. The second form of interaction, screen blast, allows software to locate the Light Pen by filling the screen with lines until the Light Pen hits one of them. The location of the Light Pen is returned in the parameters XPOS and YPOS. The LIGHT PEN activity is directed by the bits in the STATUS word. These bits are described below.



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	FOUNDPEN	This bit is set when the Refresh Processor locates a Light Pen hit while tracking or screen blasting.
14-13	not used	
12	BLAST	This bit controls the use of the Screen Blast. A Screen Blast will occur if this bit is set, tracking did not find the tracking cross and the blast test is true. To perform the blast test the Refresh Processor ands the contents of the location pointed to by MSKADR with BLASTMASK. If this results in zero, the blast test is false. The next step is to invert the sense of the blast mask test if bit 11 of the STATUS word (INVERT BLAST MASK TEST) is on.
11	INVERT BLAST TEST	This bit is used to invert the sense of blast mask test (see BLAST, bit 12).

- | | | |
|-----|-----------------------|---|
| 10 | DISPLAY ALWAYS | This bit causes the tracking cross to be displayed always. If this bit is not set, then display the tracking cross only if the track test is true. To perform the track test the Refresh Processor and's the contents of the location pointed to by MSKADR with TRACKMASK. If this results in zero, the track test is false. The next step is to invert the sense of the track bit if bit 9 of the STATUS word (INVERT TRACK TEST) is set. The Light Pen will only attempt to track when the result of this test is true. |
| 9 | INVERT TRACK TEST | This bit is used to invert the sense of the track test (see DISPLAY ALWAYS, bit 10). |
| 8 | SECOND LINE GENERATOR | This bit is clear if the Light Pen is attached to a standard Line Generator. It should be set if the Light Pen is attached to an optional second Line Generator. |
| 7-0 | SCB ADDRESS | This is the low-order eight bits of the Light Pen SCB address. The high-order bits of the SCB address are assumed to be 177400. |

The following examples all use the Command & Parameter Table of Figure 2.4-11 to illustrate the control of the Light Pen.

LG1	200	
LG2	176000	select all scopes
TRACKMASK	(see examples)	
BLASTMASK	(see examples)	
HITMASK	20	Light Pen 0
MASKPTR	177720	Standard Light Pen
HDX	0	User must write a zero to
HDY	0	initialize XPOS and YPOS
XPOS	1000	Initial X screen position
		=1000(8)
YPOS	177700	Initial screen position
		=-100(8)
STATUS	(see examples)	
CMD	15	Light Pen Command

Figure 2.4-11
Example Task Command & Parameter Table for Light Pen Control

<u>STATUS</u>	<u>BLASTMASK</u>	<u>TRACKMASK</u>	<u>Function</u>
1320	0	0	Track the Light Pen continually regardless of the TIP SWITCH. Leave the tracking cross displayed at the previous position if the pen does not point at the tracking cross.
2320	0	1	Track the Light Pen continually only if the TIP SWITCH is pressed. Leave the tracking cross displayed at the previous position if the pen does not point at the tracking cross or if the TIP SWITCH is not pressed.

PS2/MPS Reference Manual
Hardware Details

<u>STATUS</u>	<u>BLASTMASK</u>	<u>TRACKMASK</u>	<u>Function</u>
320	0	1	Track the Light Pen continually only if the TIP SWITCH is pressed. Leave the tracking cross displayed at the previous position if the TIP SWITCH is pressed and the pen does not point at the tracking cross. If the TIP SWITCH is not pressed, don't display a tracking cross.
11320	1	0	Track the Light Pen continually regardless of the TIP SWITCH. Leave the tracking cross displayed at the previous position if the pen does not point at the tracking cross and the TIP SWITCH is not pressed. However, if the TIP SWITCH is pressed then do a 'screen blast' to find the pen position (and move the tracking cross).
12320	1	1	Track the light pen continually only if the TIP SWITCH is pressed. If the pen does not point at the tracking cross and the TIP SWITCH is set, do a screen blast to find the pen position (and move the tracking cross). If the TIP SWITCH is not pressed, display the tracking cross at its previous position.
2320	0	0	Display the tracking cross. Do not track or screen blast.
320	0	0	Do not display the tracking cross and do not screen blast.

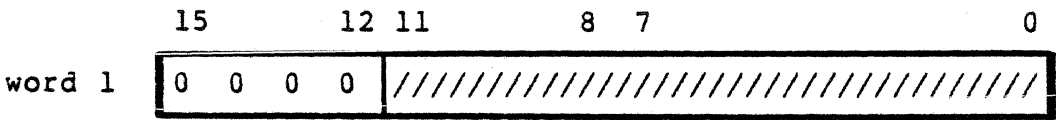
1. DEVICE CONTROL COMMAND (16, 17, 20)
Controls the graphics peripherals.

	Before	After	
DATTAB	P	P	Pointer to start of data table.
INSTAB	P	P	Pointer to start of instruction table.
LENGTH	P	P	Length of data & instruction tables.
WORD0	P	HS	
WORD1	P	HS	
WORD2	P	HS	
WORD3	P	HS	
WORD4	P	HS	
WORD5	P	HS	
WORD6	P	HS	
WORD7	P	HS	
CMD	DEVICE CONTROL	DEVICE CONTROL	

The DEVICE CONTROL COMMAND relieves the Picture Controller of much of the monitoring that is required to use the Picture System Devices such as Control Dials, Function Switches, Keyboards etc. These commands monitor the devices and cause an interrupt when a software specifiable event occurs.

The DEVICE CONTROL COMMAND has two associated tables, an instruction table and a data table, starting at the locations contained in INSTAB and DATTAB respectively. The INSTRUCTION and DATA tables are structured as parallel tables where each entry in the INSTRUCTION table has a corresponding entry in the DATA table. The length of the INSTRUCTION (or DATA) table is contained in LENGTH. Upon encountering a DEVICE CONTROL COMMAND, the Refresh Processor reads the first instruction from the INSTRUCTION table (contents of INSTAB) and performs the device control operation specified by the high-order 4 bits of the instruction word, utilizing the corresponding DATA table entry (or entries). After this operation is complete the Refresh Processor proceeds to the next instruction in the INSTRUCTION table, repeating this process until all of the instructions in the INSTRUCTION table have been executed. (The last location of the INSTRUCTION table is expected to be the address equal to INSTAB + LENGTH-1.)

The following diagrams and accompanying paragraphs detail the INSTRUCTION and DATA table entries for each available device control operation.

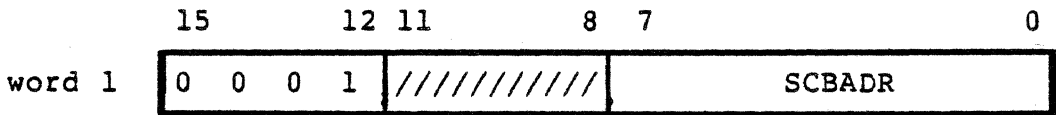


INSTRUCTION TABLE

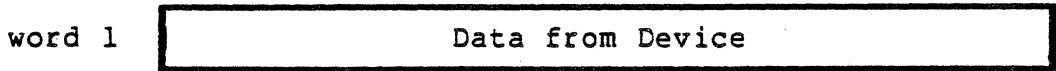


DATA TABLE

0000 Single word NOP. This instruction performs no function.



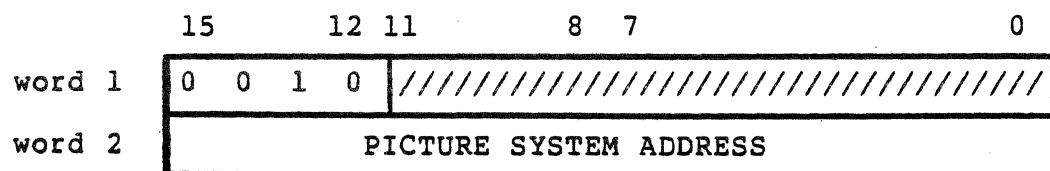
INSTRUCTION TABLE



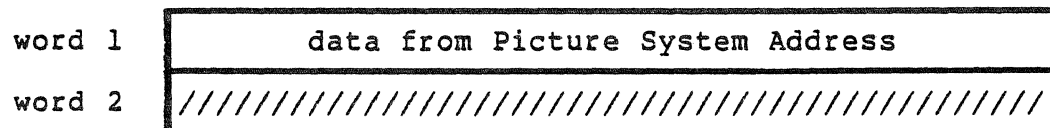
DATA TABLE

0001 Transfer data from device to DATA TABLE. The data from the address formed by adding 177400 and SCBADR are transferred into the DATA TABLE.

PS2/MPS Reference Manual
Hardware Details

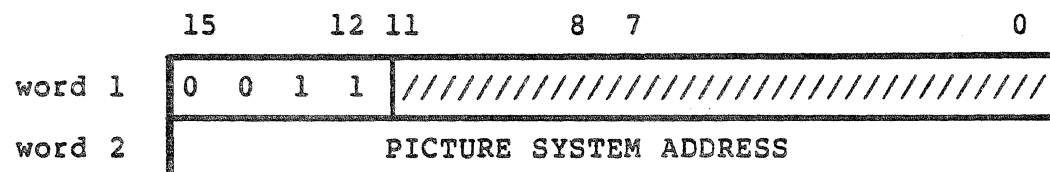


INSTRUCTION TABLE

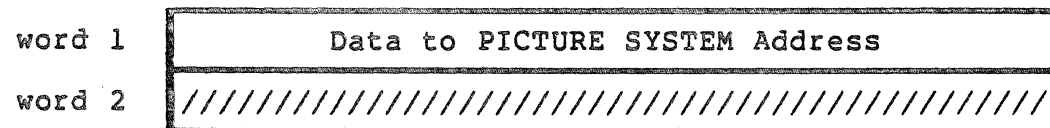


DATA TABLE

0010 Transfer data from any Picture System Address into DATA TABLE. Word 2 in the INSTRUCTION TABLE is used as an address and the data at this address are transferred into the DATA TABLE.



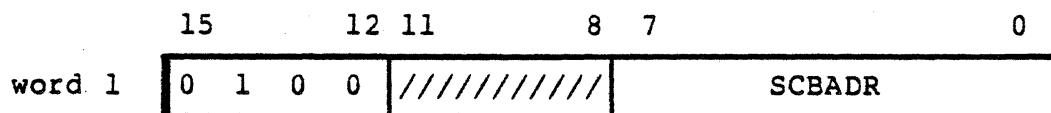
INSTRUCTION TABLE



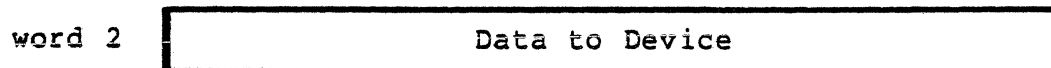
DATA TABLE

0011 Transfer from DATA TABLE to any Picture System Address. Word 2 in the INSTRUCTION TABLE is used as an address and the contents of word 1 of the DATA TABLE is transferred into this address.

PS2/MPS Reference Manual
Hardware Details

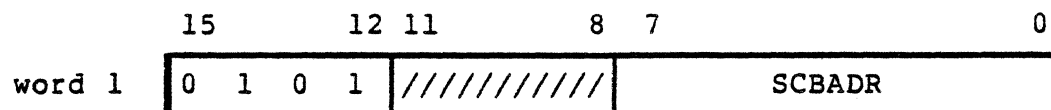


INSTRUCTION TABLE

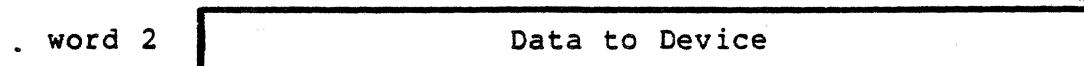


DATA TABLE

0100 Transfer from DATA TABLE to device. The contents of the DATA TABLE are moved into the address formed by adding 177400 and SCBADR.



INSTRUCTION TABLE

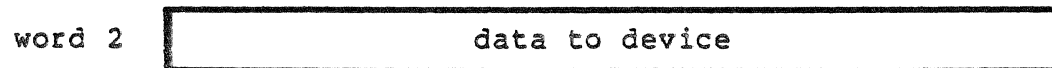


DATA TABLE

0101 Transfer data from DATA TABLE to device if last test was true. If the last test was true the contents of the DATA TABLE is moved into the address formed by adding 177400 and SCBADR.

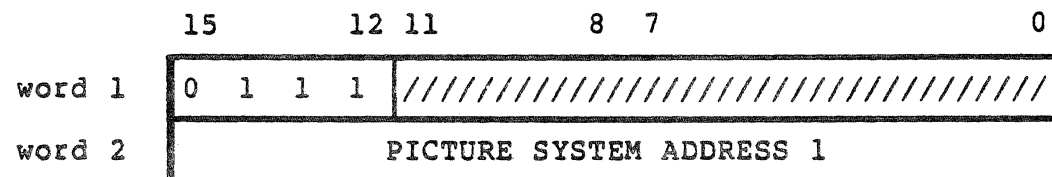


INSTRUCTION TABLE

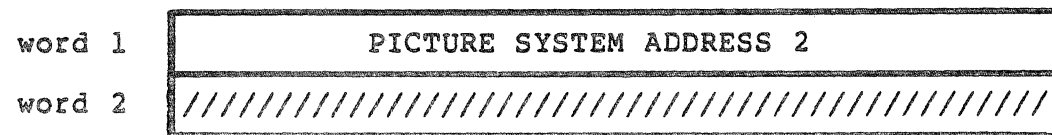


DATA TABLE

0110 Transfer data from DATA TABLE to device if the last test was false. If the last test was false, the contents of the DATA TABLE are moved into the address formed by adding 177400 and SCBADR.

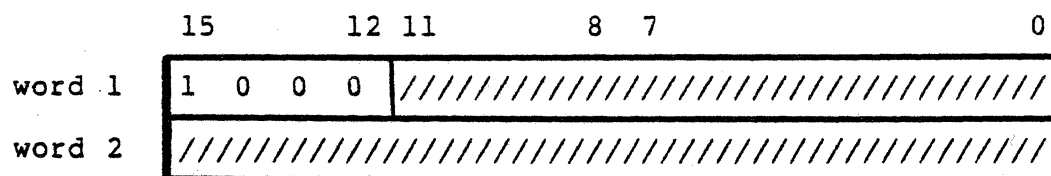


INSTRUCTION TABLE

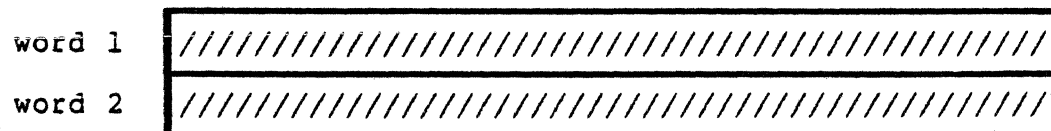


DATA TABLE

0111 Transfer data from Picture System Address 2 to Picture System Address 1. The contents of the location pointed to by word 1 of the DATA TABLE is moved into the location pointed to by word 2 of the INSTRUCTION TABLE.

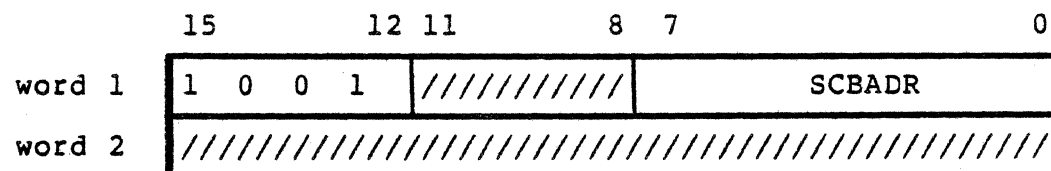


INSTRUCTION TABLE

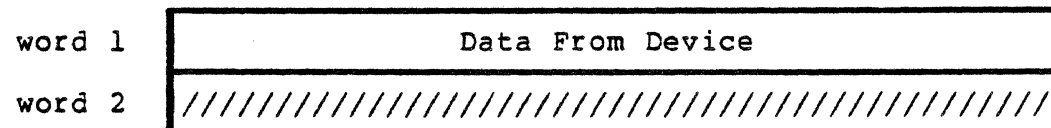


DATA TABLE

1000 Double word NOP. This command performs no function except to skip word 2 in the INSTRUCTION TABLE.



INSTRUCTION TABLE



DATA TABLE

1001 Transfer from device to DATA TABLE and skip next word. The data in the address formed by adding 177400 and SCBADR are transferred into the DATA TABLE and word 2 in the INSTRUCTION TABLE is skipped.

	15	12	11	8	7	0
word 1	1	0	1	0	////////	SCBADR
word 2	MASK					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Bits that changed to one when Test became true

DATA TABLE

1010 Read data from device and test all masked bits for change to all ones. The data in the address formed by adding 177400 and SCBADR are read and anded with the MASK. If the data from the previous cycle had some of the masked bits not equal to one and the data from the present cycle has all the mask bits equal to one, the test is true. When the test is true, the bit(s) that changed to one are set in word 2 of the DATA TABLE.

	15	12	11	8	7	0
word 1	1	0	1	1	////////	SCBADR
word 2	MASK					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Bits that changed to zero when Test became true

DATA TABLE

1011 Read data from device and test all masked bits for change to all zeros. The data in the address formed by adding 177400 and SCBADR are read and anded with the MASK. If the data from the previous cycle had some of the masked bits not equal to zero and the data from the present cycle has all the mask bits equal to zero, the test is true. When the test is true, the bit(s) that changed to zero are set in word 2 of the DATA TABLE.

	15	12	11	8	7	0
word 1	1	1	0	0	////////	SCBADR
word 2	DELTA					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Data from Device that caused Test to be true

DATA TABLE

1100 Test data change against DELTA. The data in the address formed by adding 177400 and SCBADR is compared to the contents of word 2 of the DATA TABLE and if the absolute value of the difference is greater than DELTA then the new data is transferred into both words 1 and 2 of the DATA TABLE and the test is true. If the difference is less than DELTA the test is false, and data are only transferred into word 1.

word 1	1	1	0	1	////////	SCBADR
word 2	MASK					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Bits that changed when test was true

DATA TABLE

1101 Transfer from device to DATA TABLE and test masked data for zero to non-zero change. The data in the address formed by adding 177400 and SCBADR are anded with the MASK and the result is compared with the contents of the DATA TABLE. If any bits have changed from zero to non-zero, the new data from the device are transferred and the test is true. If no bits have changed from zero to non-zero, the test is false but the new data are transferred anyway.

	15	12	11	8	7	0
word 1	1	1	1	0	////////	SCBADR
word 2	MASK					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Bits that changed when Test was true

DATA TABLE

1110 Transfer from device to DATA TABLE and test masked data for non-zero to zero change. The data in the address formed by adding 177400 and SCBADR are anded with MASK and the result is compared with the contents of the DATA TABLE. If any bits have changed from non-zero to zero, the test is true. If no bits have changed from non-zero to zero, then the test is false. The new data from the device are then transfered to the DATA TABLE.

	15	12	11	8	7	0
word 1	1	1	1	1	////////	SCBADR
word 2	MASK					

INSTRUCTION TABLE

word 1	Data from Device
word 2	Bits that changed when Test was true

DATA TABLE

1111 Transfer from device to DATA TABLE and test masked data for any change. The data in the address formed by adding 177400 and SCBADR are anded with MASK and the result is compared with the contents of DATA TABLE. If any bits have changed, the test is true. Otherwise it is false. The new data from the device are then transfered to the DATA TABLE.

Whenever a DEVICE CONTROL test (initiated by codes 1010, 1011, 1100, 1101, 1110 or 1111) is true, it sets a bit in one of 8 locations in the Command & Parameter Table (WORD 0-7). The bit is selected by subtracting the Picture System Address of the command (in the INSTRUCTION TABLE) from the address of the first location in the INSTRUCTION TABLE. If this number is greater than 128, no bit is set. Otherwise, bits 6-4 of the difference selects one of the 8 words (0 selects WORD 0, etc.) and bits 3-0 select a bit in that word (0 selects bit 0). For example, if the 40th word in the INSTRUCTION TABLE results in a true test, bit 7 of WORD 2 would be set.

There are three DEVICE CONTROL commands 16, 17, and 20.

Command 16 executes once to perform a set up for command 17. This command allows the test commands which use the previous state of a device (codes 1100, 1101, 1110, and 1111) to initialize the DATA TABLE. This command runs one cycle and does not interrupt. After it runs it changes the CMD word to 17.

Command 17 always runs completely through the instruction table. After each cycle through the instruction table, the Refresh Processor determines if any of the tests were true during that cycle. If any test was true, the Command is changed to NOP and a done interrupt sequence is initiated.

Command 20 runs continuously and neither initiates the done interrupt sequence nor changes the CMD word.

2.4.2.2.3 Refresh and Device Processor Maintenance

The Refresh Processor has a maintenance mode of operation that can be used for hardware diagnosis. The maintenance mode is entered in the following manner. After RESET, load RFDT with 177777 (octal) and then acknowledge the RP REQ bit in the System Interrupt Request Register (see Section 2.5.2). Once the Refresh Processor is in maintenance mode, eight functions are available to aid diagnosis. The functions are selected by writing a value into bits 2-0 of the RFDT register and then acknowledging the RP REQ bit in the System Interrupt Request Register. For values of RFDT(2-0) = 000-110, the operation will be performed immediately with the result available in the RFDT register when it is next read. For RFDT(2-0) = 111, the process continues indefinitely, until a RESET is issued to the Refresh Processor. The Picture Memory may then be interogated to verify the operation of the RFDT(2-0) = 111 function. The following describes the maintenance functions available.

<u>RFDT(2-0)</u>	<u>Function</u>
000	<u>RFDT Pass Through Mode</u> Data are read from the input side of RFDT and are moved through all the registers in the Refresh Processor and finally written to the output side of RFDT.
001	<u>Computed Call Check</u> Data are read from the input side of RFDT and are passed through the hardware used by the micro-program counter for computed calls and written to the output side of RFDT.
010	<u>16-bit Internal Memory Addressing Check</u> The 16 internal data registers are loaded with their individual addresses (0 in register 0, 1 in register 1, etc.) and data are read from RFDT. The low four bits of this data selects a register and the contents of that register is written into RFDT.
011	<u>Arithmetic Operations</u> The input from RFDT undergoes all the arithmetic operations used by the Refresh Processor and the result, which should be the same as the input, is written to RFDT.
100	<u>Logical Operations</u> The input from RFDT undergoes all the logical operations used by the Refresh Processor and

the result, which should be the same as the input, is written to RFDT.

- 101 Control Flip Flop Check
The various control flip flops are tested by the Refresh Processor and, if they are operating correctly, the RFDT input is transferred to RFDT output.
- 110 Shift Operations Check
The input from RFDT undergoes all the shift operations used by the Refresh Processor and should be written without modification into the output of RFDT.
- 111 Picture System Addressing and Memory Check
The contents of Picture Memory Location 0 is read, incremented and written into location 1. Location 1 is then read, incremented and written into location 2. This continues until 177377 and then starts over again. This allows the Refresh Processor hardware which reads, writes, and addresses the Picture Memory, to be tested.

2.4.3 Line Generator

The Line Generator is the unit of the Picture Generator that removes the digital data, input by the Input Controller, from the Line Generator input FIFO and converts the data into the appropriate analog signals to drive the display(s). The Line Generator removes two 16-bit words from the input FIFO and interprets the data as a 32-bit word which specifies:

- a. A Move or Draw command with X, Y and Z (intensity) information.
- b. A Status command to be interpreted by the Line Generator to change line textures, line colors, etc.
- c. A Refresh Control command for use by the Refresh Controller (PS2) or the Refresh and Device Processor (MPS) in performing refresh and segmentation operations.
- d. A Character command with four character codes to be interpreted by the Character Generator.

The data formats for each of these Line Generator words are detailed in the following sections.

PS2/MPS Reference Manual
Hardware Details

a. MOVE or DRAW Command



*=0: MOVE

*=1: DRAW



A Move command to the Line Generator causes the beam to be positioned at the X,Y position specified and at the intensity specified by the 6 bits of Z information. The beam may or may not be intensified depending upon the current status of the Line Generator.

A Draw command to the Line Generator causes a line to be drawn to the X,Y position specified and at an intensity which varies exponentially from the current beam intensity to that specified by the 6 bits of Z information. The texture of the line (i.e., long dashed, long-short dashed, etc.) is determined by the current status of the Line Generator.

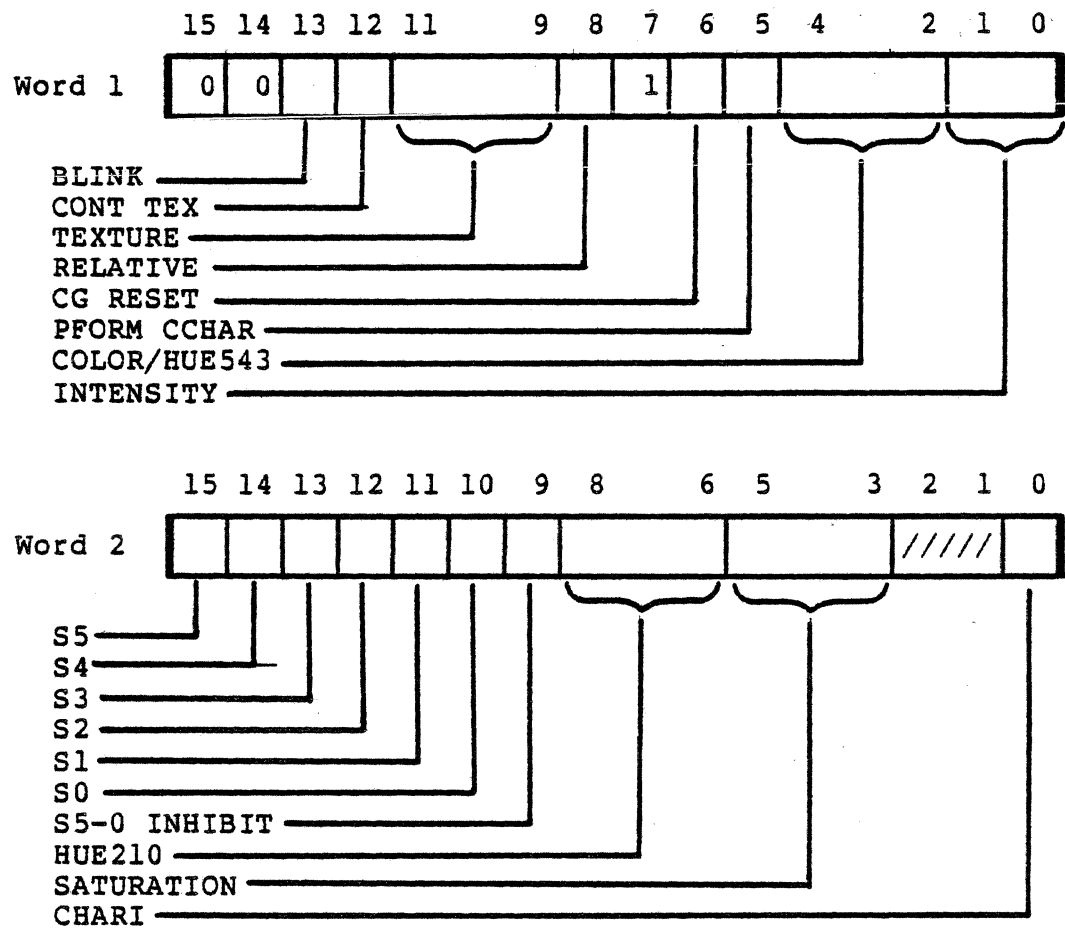
Note that if the Line Generator is currently in relative mode, as selected by the RELATIVE bit of the Status command, the X, Y and Z data are interpreted as relative data and the Move or Draw is done relative to the current position.

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 1:	15	none	This bit specifies that this is a Move or Draw command. Bit 15 must be a one.
	14	MOVE/DRAW	The state of this bit determines whether the Line Generator is to position the beam to the coordinates specified or to draw a line to the coordinates specified. MOVE/DRAW=0 for a Move; MOVE/DRAW=1 for a Draw.

PS2/MPS Reference Manual
Hardware Details

	13-12	Z5-4	These bits, in conjunction with bits 15-12 of Word 2, (Z3-0) specify the intensity of this line endpoint.
	11-0	X	These bits specify the absolute 12-bit value of the X coordinate of this line endpoint.
Word 2:	15-12	Z3-0	These bits, in conjunction with bits 13-12 of Word 1 (Z5-4), specify the intensity for this line endpoint.
	11-0	Y	These bits specify the absolute 12-bit value of the Y coordinate of this line endpoint.

b. Status Command



A Status command to the Line Generator specifies whether to put the Line Generator into blink mode, whether to display line textures as continuous textures regardless of the placement of line segments, the current line texture, the color and intensity that lines are to be displayed, what scopes are to be selected for display, and character status for the Character Generator.

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 1:	15,14	none	These bits (in conjunction with bit 7) specify that this is a Status command. Bits 15 and 14 must be zero.

PS2/MPS Reference Manual
Hardware Details

13	BLINK	This bit is set to indicate that all successive lines and characters are to blink on the display.
12	CONT TEX	CONT TEX (Continuous Texture) is set to indicate that successive textures of dashed lines are to be displayed without regard to individual line endpoints for connected line segments. When this mode is not enabled, then for all line textures the endpoints for every line segment will be displayed. This mode causes a decreased line drawing speed and should be used only when necessary.
11,10,9	TEXTURE	These bits specify how all subsequent lines are to be displayed. Lines may be drawn in any of the following textures: 000 Solid Lines All subsequent Move commands will cause the beam to be positioned to the specified X,Y coordinates, but not intensified. All subsequent Draw commands will cause a solid line to be drawn to the X,Y coordinates specified. 001 Dot Mode All subsequent Move commands will cause the beam to be positioned to the X,Y coordinates named and intensified as specified by the Z coordinate. Draw commands are not affected. 010 Short Dashed All subsequent Draw commands will cause a short dashed line to be drawn to the specified coordinates. 011 Medium Short Dashed All subsequent Draw commands

will cause a medium-short dashed line to be drawn to the specified coordinates.

100 Medium Long Dashed Lines
All subsequent Draw commands will cause a medium-long dashed line to be drawn to the specified coordinates.

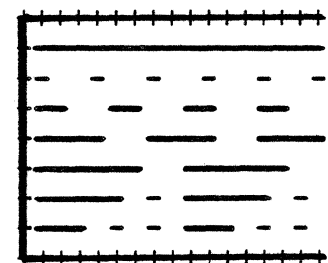
101 Long Dashed Lines
All subsequent Draw commands will cause a long dashed line to be drawn to the specified coordinates.

110 Long-Short Dashed Lines
All subsequent Draw commands will cause a line of alternating long and short dashes to be drawn to the specified coordinates.

111 Long-Short-Short Dashed Lines
All subsequent Draw commands will cause a line to be drawn to the specified coordinates which consist of a repeating long-short-short dashed line pattern.

The different line textures are shown below:

Solid
Short Dashed
Medium Short
Medium Long
Long Dashed
Long-Short
Long-Short-Short



8 RELATIVE

When RELATIVE is set, the Line Generator functions in a mode where all MOVE or DRAW commands are relative. The X and Y information is added to the current X and Y position to form the value used for the new beam position. Z is treated as absolute data.

7	none	This bit (in conjunction with bits 15 and 14) specifies that this is a Status command. Bit 7 must be a one.
6	CG RESET	When CG RESET (Character Generator Reset) is set, it reinitializes the Character Generator to its power-up state.
5	PFORM CCHAR	When PFORM CCHAR (Perform Control Character) is set, it puts the Character Generator into a mode where transfer control characters (ASCII character codes 26 and 27) are performed rather than displayed. This is equivalent to a not-show control character function. (See Section 2.4.4 for details.)
4-2	COLOR/HUE543	For the standard monochrome display and the beam penetration monitor, these bits specify the COLOR status for the scopes currently selected. The value of these bits specifies the color of all subsequent data drawn. A value of 000, 001, 010, or 011 must be used when a monochrome display is selected. Values 100-111 are used when a beam penetration color monitor is selected. Note that the INTENSITY bits must be appropriately selected for each color selection.

Color Selected

000	(Monochrome Display)
001	(Monochrome Display)
010	(Monochrome Display)
011	(Monochrome Display)
100	Red
101	Orange
110	Yellow
111	Green

For the E&S Color Display (Calligraphic Shadow Mask - CSM), these bits specify the three high-order bits of hue of all subsequent data drawn. See HUE210 of Word 2 below for further details. Note that the values for these bits specify a basic color range similar to the COLOR selection for the beam penetration display.

1-0 INTENSITY

These bits specify the relative intensity that all subsequent lines and characters are to be drawn. The gain in intensity is produced by decreasing the drawing speed of the Line Generator. This is normally used in conjunction with beam penetration monitors to produce lines of equal intensity for the different colors available and also improves the accuracy of the line end-point matching.

Intensity Selected

00	normal intensity (used for monochrome display).
01	2x normal intensity (used when selecting Green and Yellow).
10	4x normal intensity (used when selecting Orange).
11	8x normal intensity (used when selecting Red).

PS2/MPS Reference Manual
Hardware Details

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 2:	15	S5	This bit is set to indicate that the Picture Display (Scope 5) will display all data subsequently drawn. This is driven by the cable from backpanel slot 3, channel B.
	14	S4	Same as bit 15, but for Scope 4. This is driven by the cable from backpanel slot 3, channel A.
	13	S3	Same as bit 15, but for Scope 3. This is driven by the cable from backpanel slot 2, channel B.
	12	S2	Same as bit 15, but for Scope 2. This is driven by the cable from backpanel slot 2, channel A.
	11	S1	Same as bit 15, but for Scope 1. This is driven by the cable from backpanel slot 1, channel B.
	10	S0	Same as bit 15, but for Scope 0. This is driven by the cable from backpanel slot 1, channel A.
	9	S5-0 INHIBIT*	This bit is set to inhibit S5 - S0 from having any effect on the current scope selection. This allows a single scope select command to be output during a given frame without regard for any subsequent status commands. This is used in the Multi Picture System to ensure proper Picture Station scope selection.

*This feature is only available on those Picture System's that have a PS card #195211-103 (or -102) of ECO level A5 or higher.

8-6 HUE210

These bits, in conjunction with the HUE543 field in Word 1, form a 6-bit value which is used to select the HUE of all subsequent data drawn for the E&S Color Display (Calligraphic Shadow Mask - CSM). When the SATURATION field (bits 5-3) = 7, the HUE value then selects a color which is dependent upon the value specified. As HUE=0-40, the color will range from green (HUE=0) to cyan (HUE=10) to blue (HUE=20) to magenta (HUE=30) to red (HUE=40). As HUE=40-77, the color will range from red (HUE=40) to yellow (HUE=60) to almost green (HUE=77). As the SATURATION is reduced (SATURATION field = 6-0), each color will become more pastel. For HUE=0 and SATURATION=0, white should be displayed.

5-3 SATURATION

These bits specify a value which is used to specify the saturation of color specified by the HUE value for the E&S Color Display. As this value is reduced from the maximum saturation level (7), each color will become more pastel. For HUE=0 and SATURATION=0, white should be displayed.

2-1 not used

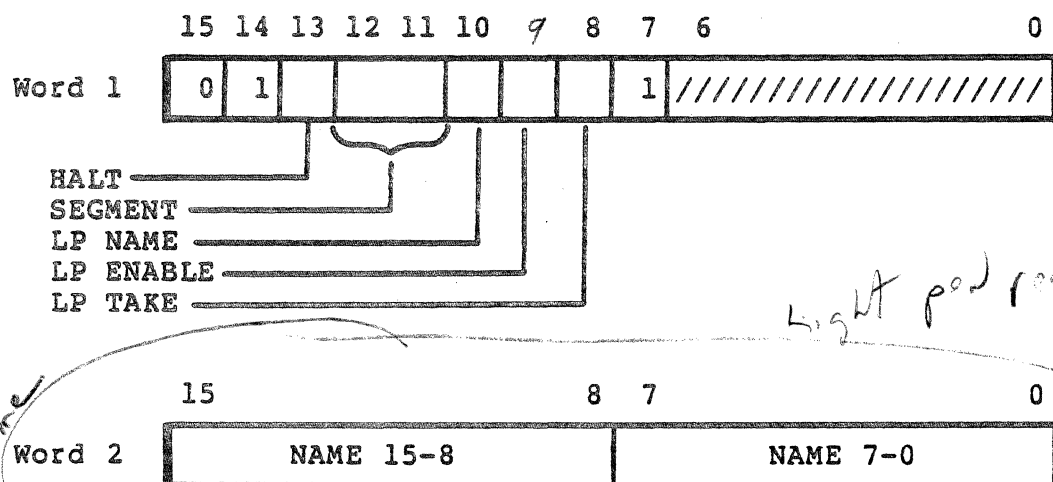
0 CHARI

CHARI (Character Intensity) is set to specify that the intensity characters are to be displayed is to be taken from the Intensity field of the Character Generator Font Parameter Stack (see Section 2.4.4.4 for specific details). When CHARI is not set, the intensity characters are displayed is the intensity associated with the last Move or Draw command which positioned for the character string.

c. Refresh Control Command

The Refresh Control Command directs the device which is controlling the refresh of the display to perform a control function. The format and function of the Refresh Control Command is different for the Picture System 2 and the Multi Picture System. This is because the only hardware difference between the two systems is that a Picture System 2 contains a Refresh Controller and the Multi Picture System contains a more intelligent and powerful Refresh and Device Processor. However, the Refresh Control Command is similar for both devices. The Refresh Control Commands for both systems are detailed below.

Picture System 2 Refresh Control Command



A Refresh Control command causes the function specified by bits 13-11 to be performed. If no function is enabled (bits 13-11=0), these two words will be passed onto the Line Generator Input FIFO and the next two 16-bit words fetched from PSMEM. The Line Generator performs no function upon receiving a Refresh Control command (unless a Light Pen operation is specified as part of the Refresh Control command).

PS2/MPS Reference Manual
Hardware Details

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 1:	15-14	none	These bits (in conjunction with bit 7) specify that this is a Refresh Controller command. Bit 15 must be a zero and bit 14 must be a one.
	13	HALT	This bit is used to indicate the end of a refresh segment and that all further input to the Line Generator is to cease. The encounter of a HALT command by the Refresh Controller is equivalent to the RFAIA=RFAIL.
	12-11	SEGMENT	These bits are set to specify a segment name and how that segment name is to be used. The bit combinations and interpretations are as follows:
		00	No function.
		01	SEGMENT Jump. This command causes the Refresh Controller to jump to the absolute address in PSMEM specified by Word 2. This is done by loading the RFAIA with the contents of Word 2. It should be noted that when a SEGMENT Jump command is encountered by the Refresh Controller when WRITE-BACK mode is enabled, the Jump will <u>not</u> be written back; however, the Jump will be performed, thereby compacting the physically disjoint data segments into a contiguous segment.
		10	SEGMENT Name. This command causes the Refresh Controller to load the segment name specified by Word 2 into the RFCSN register. If the SEARCH bit in the RFSR is set, the Refresh Controller compares the RFCSN with the RFSN register which was loaded under program control. If the RFCSN is

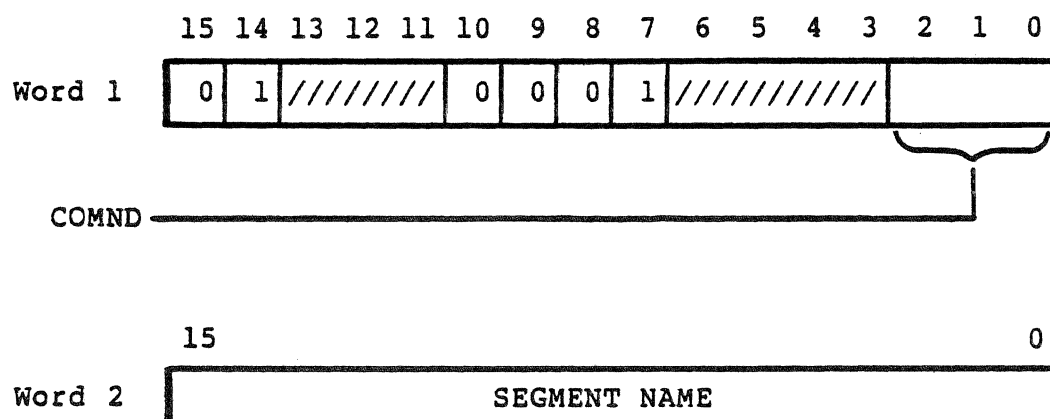
equivalent to RFSN, a MATCH is declared by the Refresh Controller and the MATCH REQ bit set in the System Interrupt Request register.

11		SEGMENT Name-Blanked. This command functions in the same manner the SEGMENT Name (10) command functions, but sets the Refresh Controller to a mode whereby this command and all subsequent data will not be passed on to the Line Generator for display until the next SEGMENT Name command is encountered. This allows segments to be maintained in the refresh buffer but not displayed.
10	LP NAME	This bit is set to cause the segment name specified by Word 2 to be loaded into the Light Pen Segment Name register (LPSN, see Section 2.6.6). This bit is valid only if the LP TAKE bit (8) is also set.
9	LP ENABLE	This bit is set to enable Light Pen sensitivity. This allows a program to make only certain portions of a picture visible to the light pen. This bit is valid only if the LP TAKE bit (8) is also set.
8	LP TAKE	This bit is set to allow bits 9 and 10 to have effect on the current status of the Light Pen. If this bit is not set, then LP NAME and LP ENABLE are ignored.
7	none	This bit (in conjunction with bits 15 and 14) specifies that this is a Refresh Controller command. Bit 7 must be a one.
6-0	unused	

PS2/MPS Reference Manual
Hardware Details

Word 2:	15-8	NAME 15-8	These bits constitute the 8 most-significant bits of the Segment Name. The SEARCH MODE bits in the RFSR register may be set so that in searching for a segment name these bits are either masked off or included in the comparison.
	7-0	NAME 7-0	These bits constitute the 8 least-significant bits of the segment name. The SEARCH MODE bits in the RFSR register may be set so that in searching for a segment name the bits are either masked off or included in the comparison.

Multi Picture System Refresh Control Command



A Refresh Control Command causes the Refresh and Device Processor to perform the function specified by the COMND field (bits 2-0) of Word 1. When a segment is specified as Light Pen sensitive (COMND = 101 or 111), a Picture System 2 Refresh Control Command with the appropriate bits (i.e., LP NAME, LP ENABLE, LP TAKE) set is sent to the Line Generator Input FIFO to perform the Light Pen control function.

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 1:	15-14	none	These bits in conjunction with bit 7 specify that this is a Refresh Control Command. Bit 15 must be a zero and bit 14 must be a one.
	13-11	not used	
	10-8	none	These are the Light Pen control bits. These must be zero for proper Light Pen operation.
	7	none	This bit in conjunction with bits 15 and 14 specifies that this is a Refresh Control Command. Bit 7 must be a one.
	6-3	not used	
	2-0	COMND	These bits specify the eight commands that are used to control the refreshing of Picture

Memory. The bit combinations and interpretations are as follows:

000	No Operation.
001	No Operation.
010	<u>HALT</u> This command indicates that the Refresh and Device Processor has reached the end of the display list and requires no further input to the Line Generator Input Controller.
011	<u>JUMP</u> This command causes the Refresh and Device Processor to jump to the absolute address in PSMEM specified by Word 2. Refresh will continue from that point.
1xx	<u>SEGMENT</u> The four commands with bit 2 set are the segment name commands and the Refresh and Device Processor uses word 2 as a segment name. Segment names occur in the display list in pairs. The transformed display data which resides between a pair of segment names forms a "segment". A segment may contain Line Generator Move, Draw, Status, Refresh Control or Character Commands. A segment may also contain other (nested) segments.

Bit 0 of the COMND field controls whether or not a segment is to be Light Pen sensitive. If this bit is set in the first name of a segment, the contents of that segment are able to be detected by a Light Pen. When an outer segment is not Light Pen sensitive, an inner segment may be Light Pen sensitive if bit 0 of it's COMND field is set.

Bit 1 of the COMND field controls whether a segment is

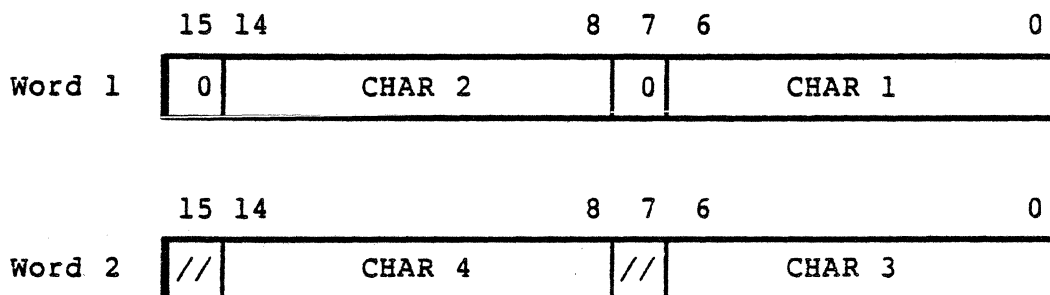
blanked (not drawn on the screen) or unblanked. If this bit is set in the first name of a segment, the Line Generator is disabled until the second occurrence of the name. Blanking an outer segment blanks all inner segments while unblanking an outer segment allows the inner segments to be blanked or unblanked according to their individual COMND fields.

Word 2: 15-0

SEGMENT NAME

When the COMND field of Word 1 is a No Operation command (000 and 001) or a Halt command (010), this word is unused. When the COMND field of Word 1 is a JUMP command (011), this word contains the associated 16-bit absolute address in Picture Memory. When the COMND field of Word 1 is a SEGMENT command (1xx), this word contains a 16-bit segment name. The segment names are used to locate and control the data in the transformed display list.

d. Character Command



A Character command to the Line Generator causes four character codes to be passed to the Character Generator for interpretation into strokes which are passed back to the Line Generator for display.

	<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
Word 1:	15	none	This bit (in conjunction with bit 7) specifies that this is a Character command. Bit 15 must be a zero.
	14-8	CHAR 2	These bits specify the character to be displayed at the current beam position.
	7	none	This bit (in conjunction with bit 15) specifies that this is a Character command. Bit 7 must be a zero.
	6-0	CHAR 1	These bits specify the character to be displayed at the current beam position.

PS2/MPS Reference Manual
Hardware Details

Word 2:	15	none	This bit is or'ed with the least-significant of the Character Style bits and used with the seven bits of character code to specify the beginning location in the selected Character Memory (PROM or RAM) that the character stroke command begins.
	14-8	CHAR 4	These bits specify the character to be displayed at the current beam position.
	7	none	This bit is or'ed with the least-significant of the Character Style bits and used with the seven bits of character code to specify the beginning location in the selected Character Memory (PROM or RAM) that the character stroke command begins.
	6-0	CHAR 3	These bits specify the character to be displayed at the current beam position.

2.4.4 Character Generator

The Character Generator is the unit of the Picture Generator that accepts character codes from the Line Generator and interprets these codes, producing strokes (i.e., Line Generator Move and Draw commands). These strokes are then channeled back to the Line Generator for display. There are three basic units of the Character Generator:

1. The Character Control
2. The Character Memory
3. The Character Multiplier

These units function together to cause any of the 95 ASCII characters to be drawn in any of eight standard sizes for regular or italicized characters. The Character Generator may also be programmed to provide many additional sizes and orientations. These units may also cause alternate character fonts and control information to be loaded. The following sections describe the functions of the Character Generator and detail the individual units.

2.4.4.1 Standard Character Sizes and Definitions

The standard Character Generator comes with the ROM Character Memory programmed to interpret the 128 ASCII character set shown in Table 2.4-3 and display the 95 displayable ASCII character subset shown in Table 2.4-4. The ROM scaling coefficients provide eight different sizes of characters in horizontal orientation and eight different sizes in horizontal italicized orientation. The character sizes provided are:

- 0. .36 cm (.14 inches, 98 characters across full screen)
- 1. .08 cm (.03 inches, 342 characters across full screen)
- 2. .15 cm (.06 inches, 228 characters across full screen)
- 3. .25 cm (.10 inches, 147 characters across full screen)
- 4. .40 cm (.16 inches, 81 characters across full screen)
- 5. .68 cm (.27 inches, 49 characters across full screen)
- 6. 1.14 cm (.45 inches, 30 characters across full screen)
- 7. 1.88 cm (.74 inches, 18 characters across full screen)

These sizes give the approximate height of a capital letter (A-Z) based upon a 28.6 x 28.6 cm (11.2 x 11.2 inch) screen viewing area. The Character Generator is initialized to size 0 by RESET. It should be noted that subscript and superscript characters are available only in sizes 3-8. Subscript or superscript character codes (30-33) used when size 1 or 2 is selected will result in invalid character size selection.

Control of the Character Generator is achieved by a set of control characters which are used to regulate character sizes, subscript, superscript, italics, and character line positioning. Control characters and their functions are detailed in Table 2.4-5.

Table 2.4-3
128 ASCII Character Set

ASCII CODE	CHARACTER	ASCII CODE	CHARACTER	ASCII CODE	CHARACTER
000	P (NULL)	040	space	120	P
001	A (SOH)	041	!	121	Q
002	B (STX)	042	"	122	R
003	C (ETX)	043	#	123	S
004	D (EOT)	044	\$	124	T
005	E (ENQ)	045	%	125	U
006	F (ACK)	046	&	126	V
007	G (BEL)	047	'	127	W
010	H (BS)	050	(	130	X
011	I (HT)	051	)	131	Y
012	J (LF)	052	*	132	Z
013	K (VT)	053	+	133	[
014	L (FF)	054	,	134	\
015	M (CR)	055	-	135	]
016	N (SO)	056	.	136	^
017	O (SI)	057	/	137	`
020	P (DLE)	060	0	140	~
021	Q (DC1)	061	1	141	a
022	R (DC2)	062	2	142	b
023	S (DC3)	063	3	143	c
024	T (DC4)	064	4	144	d
025	U (NAK)	065	5	145	e
026	V (SYN)	066	6	146	f
027	W (ETB)	067	7	147	g
030	X (CAN)	070	8	150	h
031	Y (EM)	071	9	151	i
032	Z (SUB)	072	:	152	j
033	K (ESC)	073	;	153	k
034	L (FS)	074	<	154	l
035	M (GS)	075	=	155	m
036	N (PS)	076	>	156	n
037	O (US)	077	?	157	o
		100	@	160	p
		101	A	161	q
		102	B	162	r
		103	C	163	s
		104	D	164	t
		105	E	165	u
		106	F	166	v
		107	G	167	w
		110	H	170	x
		111	I	171	y
		112	J	172	z
		113	K	173	{
		114	L	174	
		115	M	175	}
		116	N	176	~
		117	O	177	del

Table 2.4-4
95 Displayable ASCII Characters

ASCII CODE	CHARACTER	ASCII CODE	CHARACTER
040	space	120	P
041	!	121	Q
042	"	122	R
043	#	123	S
044	\$	124	T
045	%	125	U
046	&	126	V
047	'	127	W
050	(	130	X
051	)	131	Y
052	*	132	Z
053	+	133	[
054	,	134	\
055	-	135	]
056	.	136	^
057	/	137	_
060	0	140	
061	1	141	a
062	2	142	b
063	3	143	c
064	4	144	d
065	5	145	e
066	6	146	f
067	7	147	g
070	8	150	h
071	9	151	i
072	:	152	j
073	;	153	k
074	<	154	l
075	=	155	m
076	>	156	n
077	?	157	o
100	@	160	p
101	A	161	q
102	B	162	r
103	C	163	s
104	D	164	t
105	E	165	u
106	F	166	v
107	G	167	w
110	H	170	x
111	I	171	y
112	J	172	z
113	K	173	{
114	L	174	
115	M	175	}
116	N	176	~
117	O	177	del

Table 2.4-5
Control Character Codes and Functions

<u>ASCII CODE</u>	<u>MNEMONIC</u>	<u>FUNCTION</u>
002	STX	Start of Text. Causes the current character position to be updated as the current top and left margin position.
003	LFT (ETX)	Push-Load Font Stack. Causes the six low-order bits of the next two characters received to be assembled into a 12-bit number. The number is pushed onto the Font Stack as the Intensity, Style and Coefficient parameters. The coefficient offset is set to zero. (See Section 2.4.4.4.) 2-196
004	PFT (EOT)	Pop the Font Stack. Causes the Font Parameters to be set to those parameters specified in the last Push-Load Font Stack (LFT) instruction. (See Section 2.4.4.4.)
010	BS	Backspace. Causes the current character position to be updated to the previous character position on the line.
011	HT	Horizontal Tab. Causes the current character position to be updated to the next tab stop on the line. Tab stops are defined by default to be at the following space positions: 8,16,24,32,40,48,....
012	LF	Line Feed. Causes the current character position to be updated to the next line without modifying the relative character position within the lines.
013	VT	Vertical Tab. Causes eight line feeds to be performed.
014	FF	Form Feed. Causes the current

		character position to be updated to the current top and left margin position.
015	CR	Carriage Return. Causes the current character position to be updated to the current left margin position.
026	SXF (SYN)	Start Transfer. Causes the subsequent characters received (until an End Transfer, 027) to be used as special data for the Character Generator rather than interpreted as character codes. This mode is used to load the RAM Character Memory, and to perform other initialization functions. This operation is <u>only</u> performed if the current <u>Line</u> Generator Status was set with the PFORM CCHAR bit set (see Section 2.4.3b).
027	EXF (ETB)	End Transfer. Causes the mode of the Character Generator to be reset so that all subsequent characters received will be interpreted as characters. Used to reset the mode enabled by SXF (026). This operation is <u>only</u> performed if the current <u>Line</u> Generator Status was set with the PFORM CCHAR bit set (see Section 2.4.3b).
030	SUP (CAN)	Superscript. Causes the current character position to be updated to the raised superscript position and the current character size to be reduced to the next smaller size.
031	RSP (EM)	Reset Superscript. Causes the current character position to be updated to the current line position and the current character size to be increased to the next larger size.
032	SUB (SUB)	Subscript. Causes the current character position to be up-

PS2/MPS Reference Manual
Hardware Details

dated to the lowered subscript position and the current character size to be reduced to the next smaller size.

033	RSB (ESC)	Reset Subscript. Causes the current character position to be updated to the current line position and the current character size to be increased to the next larger size.
034	SIT (FS)	Set Italics. Causes the current size characters to be placed into italics mode whereby all subsequently displayed characters will be slanted to the right.
035	RIT (GS)	Reset Italics. Causes the italics mode to be reset so that characters are displayed in their normal orientation.
036	CUR (RS)	Cursor. Causes a blinking underline to be displayed at the current character position with the current character position unmodified.

2.4.4.2 The Character Control

The Character Control is the unit of the Character Generator that interprets the character codes received from the Line Generator. The Character Control provides for four selectable character styles of which two standard styles are provided. One or two programmable styles may be implemented by the user in the RAM Character Memory and the RAM Font Parameter Stack (see Section 2.4.4.3 for further details). For any style, the interpretation of the characters entails executing the commands contained in the Character Memory for each character, scaling the generated strokes by the Character Multiplier and channelling the scaled strokes to the Line Generator for display.

The Character Generator input and status is controlled by Status commands received by the Line Generator. (See Section 2.4.3b for details.)

2.4.4.3 The Character Memory

The Character Memory consists of two 1024 12-bit word memories which contain the information defining strokes for each of the characters which may be interpreted by the Character Generator. One memory is a 1024-word PROM which has been microcoded to provide the characters and functions described in Section 2.4.4.1, and one memory is a 1024-word Random Access Memory (RAM) which may be loaded with one or two user-defined font descriptions or special symbols.

The Character Memory contains character generation instructions (one per 12-bit word) which define the strokes to be produced for each character. When a character is received by the character control, the character code, in conjunction with the currently-selected font, is used to address the PROM (or RAM). The character generation instructions are then executed until a character termination instruction is executed which causes the Character Control to fetch the next character. Each character is defined as a series of line segments (Moves and Draws) on a 16x16 graph as shown in Figure 2.4-12.

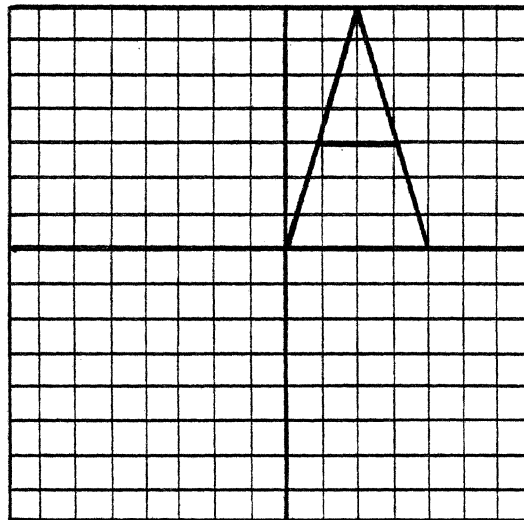
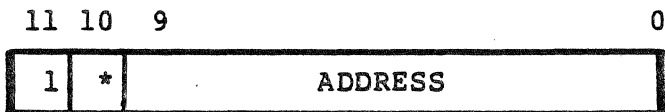


Figure 2.4-12
Character Generation Definition Graph

The instructions which are used for the character definitions are described as follows:

a. JMS



*=0: ADDRESS is within PROM

*=1: ADDRESS is within RAM

The JMS (JuMp Subroutine) instruction causes the next character generation instruction to be taken from the ADDRESS specified (PROM or RAM as indicated by bit 10). The first 128 (or 256) words of each Character Memory contains JMS instructions. When a character code is input, the code (0-127 or 0-255) is used as the starting address in the character memory for the generation of the strokes for the character. This starting address must contain a JMS instruction whose associated ADDRESS is the starting location of the actual stroke generating instruction. The initial JMS (in locations 0-127 or 0-255) is used as a simple jump in this instance rather than a subroutine jump.

b. REMOVE



E=0: Not End

E=1: End

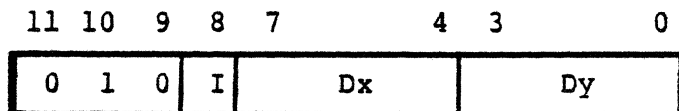
AD=0: Do Not Add Displacement to Dx and Dy

AD=1: Add Displacement to Dx and Dy

The RMOVE instruction causes the current character position to be updated to the current position plus the Delta x and y values ($-9 < Dx, Dy < 8$). If E (End, bit 9) is set, then the current character interpretation is ended. If AD (Add Displacement, bit 8) is set, the current contents of the X and Y Displacement registers (see Section 2.4.4.4) are added to the Dx and Dy values to form the final position for the next character. The RMOVE instruction (with E=1) is used only after all other strokes of the character are processed to perform the final positioning for the next character to be displayed.

Note that when the "Add Displacement" operation is performed, the Displacement values are added directly to internal position registers. The resulting position change on the screen is unaffected by the Character Multiplier parameters (see Section 2.4.4.4).

c. CHDRAW/CHMOVE

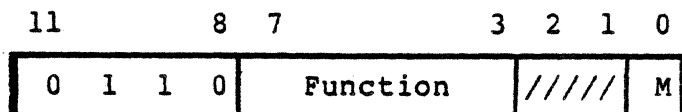


I=0: CHMOVE (Do Not Intensify)

I=1: CHDRAW (Intensify)

The CHDRAW/CHMOVE instruction is used to specify the individual strokes which compose the character to be generated. This CHMOVE differs from the RMOVE instruction in that the current character position is not updated and is therefore used only for intra-character positioning. During CHDRAW/CHMOVE instructions, the beam position is changed by an analog integrator in the line-generating hardware. The position is not "fixed" to a digitally-controlled position again until an RMOVE instruction is executed. For this reason, more than one RMOVE instruction may be required in a very complex character description to avoid "drift". If I (Intensify, bit 8) is set, a stroke is produced from the current beam position to the relative Delta x and y position ($-9 < Dx, Dy < 8$). The character definitions consist mostly of CHDRAW/CHMOVE instructions.

d. CONTROL



Function= Function to be performed (see below)

M= Modifier - Function dependent (see below)

The CONTROL command is used to perform the miscellaneous functions required for the Character Generator to be the powerful and versatile device that it is. The Function field is used to name the particular function to be performed. The Modifier bit is generally used to indicate that the current character generation is complete (End), but may also be used to specify other modification to the function to be performed. These fields are detailed below:

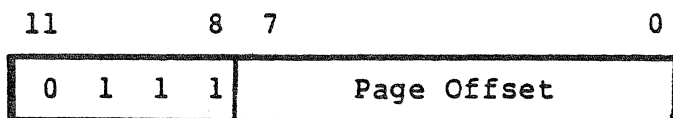
<u>Function</u>	<u>Description</u>
00	Reset Character generator. If M=1, then End.
01	Null. Character Memory no-op. If M=1, then End.
02	Transfer LIMIT to x,y position registers. LIMIT is used as a temporary storage for x,y position. If M=1, then End.
03	Save current character position as left and top margin position (or HOME position). If M=1, then End.
04	Set the current x,y position to the saved HOME position. If M=1, then set the x,y position to the saved carriage return position (CR position) and End.
05	Save current character position as the carriage return position (CR position). If M=1, then End.
06	Save the current x,y position in LIMIT and then set the x,y position to the saved CR position. This instruction, with M=1, must only be used for a carriage return character. For other characters (i.e., the TAB and LINE FEED characters), then M=0. Also, if M=1, then End.
07	Return from subroutine (JMS). Note: subroutines may only be called one level deep. An attempt to JMS within a subroutine will cause the original return address to be destroyed.
10	Null. Character Memory no-op. If M=1, then End.
11	Test REPEAT counter and skip next instruction if REPEAT=-1; otherwise, increment REPEAT and proceed to the next instruction.
12	Skip next instruction if PFORM CCHAR mode is not set.
13	Skip next instruction if the current x,y position does not exceed LIMIT.

- 14 Skip next instruction and increment REPEAT until REPEAT=-1. One instruction will be skipped for each incrementation of REPEAT.
- 15 Load REPEAT counter with the next character received by the Character Generator.
- 16-17 Load REPEAT counter with the contents of bits 3=0 of this instruction.
- 20 Push the contents of the next two character codes received by the Character Generator onto the Font Parameter Stack and zero the coefficient offset. If M=1, then End.
- 21 Skip the next instruction in the Character Memory, but push the contents of the skipped word onto the Font Parameter Stack and zero the coefficient offset. If M=1, then End.
- 22 For M=0: Pop the top element of the Font Parameter Stack off the stack. (This instruction, with the M bit set to 1, must only appear in the set of instructions which implements the LINE FEED function. Specifically, it must follow the last RMOVE instruction).
- 23 Increment the Font Parameter Stack stack pointer. If M=1, then End.
- 24 Increment the Font Parameter Stack pointer, copy the Character Style and Coefficient Address fields from the previous entry in the stack, and zero the Coefficient Offset field. (This instruction must only appear in the set of instructions which implements the Line Feed function. Specifically, it must precede the first RMOVE instruction).
- 25 Set blink mode so that all subsequent character strokes generated until the next RMOVE instruction will blink during display. If M=1, then End.
- 26 Undefined

27 For M=0: The Character Generator is set to a mode where all subsequent characters received will be taken as special data for the Character Generator. This mode is set to allow the RAM Character Memory, coefficients, internal registers, etc. to be loaded. If PFORM CCHAR is not set, then End.
For M=1: Character Memory no-op.

30-37 Add the value specified by bits 5-0 of this instruction to the coefficient offset of the Font Parameter Stack.

e. JMP



The JMP (JuMp Page) instruction is used to transfer the control of the character generation instructions to the location offset specified within the current page (Page Offset). Each Character Memory (PROM and RAM) is considered to be four 256-word pages (0-255, 256-511, 512-767, 768-1023). The Page Offset in the JMP instruction specifies the offset from the beginning of the current page. It is not possible to cross page boundaries using the JMP instruction.

2.4.4.4 The Character Multiplier

The Character Multiplier is the unit of the Character Generator that scales and/or rotates the RMOVE and CHDRAW/CHMOVE instruction data taken from the Character Memory. There are three basic units of the Character Multiplier:

1. PROM storage for 16 sets of coefficients.
2. RAM storage for 32 sets of coefficients.
3. A 16-level Font Parameter Stack

The coefficient PROM and RAM are used to contain values by which all character RMOVE and CHDRAW/CHMOVE instructions are multiplied before they are passed to the Line Generator for display. The values in these memories are called sets of coefficients because they represent the sets of coefficients in the following equations. The coefficients may be conveniently represented as elements of a 2x2 matrix as follows:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A & C \\ B & D \end{bmatrix} = \begin{bmatrix} x' & y' \end{bmatrix}$$

By varying these coefficients, the characters may be scaled to be at any angle of orientation (i.e., horizontal, vertical, etc.), at any degree of slant (for variable italic emphasis) and at up to 64 different sizes (actually 4096 since there are 64 x 64 values available). The following examples illustrate use of the coefficients to produce a) horizontal character orientation at the largest size, b) vertical (counter-clockwise) orientation at the smallest size, and c) italicized characters at mid-range size:

a) A=63, B=0, C=0, D=63

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 63 & 0 \\ 0 & 63 \end{bmatrix} = \begin{bmatrix} 63x & 63y \end{bmatrix}$$

b) A=0, B=-1, C=1, D=0

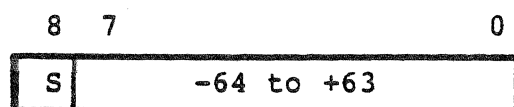
$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -1y & 1x \end{bmatrix}$$

c) A=31, B=15, C=0, D=31

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 31 & 0 \\ 15 & 31 \end{bmatrix} = \begin{bmatrix} 31x+15y & 31y \end{bmatrix}$$

The coefficient PROM and RAM are memories containing sets of four 9-bit values which are taken as the A, B, C and D coefficients. The PROM contains 16 sets of coefficients, addressed 0-17. The RAM contains 32 sets of coefficients, addressed 40-77.

Each 9-bit value is interpreted as 1 bit of character speed control, 2 bits of sign and 6 bits of data. It should be noted that the high-order sign bit is not used. This provides for up to 64 different character sizes available.



S=Speed Control

The character speed control bits are used to control the actual speed (and therefore size) that character strokes are drawn. The speed control bit (S) must be the same for the A and C coefficients and the B and D coefficients. Thus, these speed control bits form a two-bit field which specifies the following:

<u>B/D</u>	<u>A/C</u>	<u>Meaning</u>
0	0	Normal character speed and size. All characters are drawn with strokes of constant speed at the fastest rate possible. Used for large-sized characters.
0	1	Half normal speed and size. All characters are drawn with strokes of constant speed at half the fastest rate possible. Used for medium-sized characters.
1	0	Quarter normal speed and size. All characters are drawn with strokes of constant speed at one-quarter the fastest rate possible. Used for small-sized characters.
1	1	Eighth normal speed and size. All characters are drawn with strokes of constant speed at one-eighth the fastest rate possible. Used for very small-sized characters.

The speed control bits provide for the finer resolution and size selection required to accurately represent small characters. It should be noted that even though characters are drawn at a slower speed, the hardware compensates to retain the same relative intensity.

Table 2.4-6
Standard PROM Coefficients

PROM Addr	Char Size	Coefficients*				Speed Bits		Speed Drawn	Scale & Orientation
		A	B	C	D	B/D	A/C		
0	0	34	0	0	34	1	0	1/4	7, Horizontal, Normal
1	1	20	0	0	20	1	1	1/8	2, Horizontal, Normal
2	2	30	0	0	30	1	1	1/8	3, Horizontal, Normal
3	3	24	0	0	24	1	0	1/4	5, Horizontal, Normal
4	4	42	0	0	42	1	0	1/4	8.5, Horizontal, Normal
5	5	70	0	0	70	1	0	1/4	14, Horizontal, Normal
6	6	56	0	0	56	0	1	1/2	23, Horizontal, Normal
7	7	46	0	0	46	0	0	full	38, Horizontal, Normal
10	0	34	16	0	34	1	0	1/4	7, Horizontal, Italic
11	1	20	10	0	20	1	1	1/8	2, Horizontal, Italic
12	2	30	14	0	30	1	1	1/8	3, Horizontal, Italic
13	3	24	12	0	24	1	0	1/4	5, Horizontal, Italic
14	4	42	21	0	42	1	0	1/4	8.5, Horizontal, Italic
15	5	70	34	0	70	1	0	1/4	14, Horizontal, Italic
16	6	56	27	0	56	0	1	1/2	23, Horizontal, Italic
17	7	46	23	0	46	0	0	full	38, Horizontal, Italic

*The coefficients are octal values, but the effective scales are decimal.

The Font Parameter Stack is a 16-level stack which holds the parameters defining the current character font. These parameters are:

1. Character Intensity
2. Character Style
3. Coefficient Address
4. Coefficient Offset

With the Z-Depth Cue feature, the character intensity is specified by 4 bits providing 16 distinct levels of intensity. Characters are normally at the intensity of the last Move or Draw command interpreted by the Line Generator. If the character intensity is to be selected independent of the Move or Draw commands to the Line Generator, the Line Generator status must be updated with the CHAR1 bit set indicating that the character intensity information should be taken from the Font Parameter Stack (see Section 2.4.3 for specific details of the Line Generator status format).

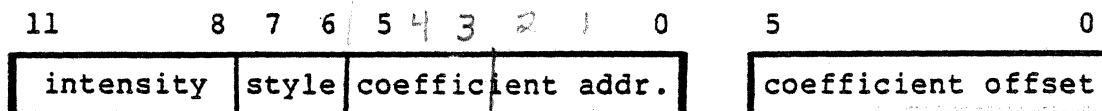
The character style is specified by 2 bits providing four selectable character styles in the PROM and RAM Character Memories (two styles in each). The most-significant style bit is used to indicate that the PROM (0) or RAM (1) is to be selected. The least-significant of the style bits is used, along with the eight bits of the actual character code, to specify the beginning location in the selected Character Memory (PROM or RAM) of the character stroke commands (see Section 2.4.3d).

The coefficient address is specified by 6 bits providing 64 addressable coefficient locations--16 in the coefficient PROM and 32 in the coefficient RAM.

The coefficient offset is specified by 6 bits and is added to the coefficient address parameter to form the final coefficient address for this Font Parameter Stack entry. The coefficient offset allows indexing within a basic set of coefficients by adding to or subtracting from the coefficient offset value.

PS2/MPS Reference Manual
Hardware Details

These parameters, described above, define the current character font. The top element of the Font Parameter Stack is always used to select the current character intensity, character style, and coefficients. The CONTROL instructions, previously defined in the Character Memory (Section 2.4.4.3), are used to load and manipulate the contents of the Font Parameter Stack. Each element of the Font Parameter Stack is of the following form:



2.4.4.5 Detailed Programming of the Character Generator

The Character Generator has many capabilities which are not utilized when the standard PROM Character and Coefficient Memories are used. Among these capabilities are the ability to load the RAM Character and Coefficient Memories, load the inter-character and inter-line spacing registers, to embed comments (i.e., non-displayable character strings) with a string of characters, and to dispatch directly to a location within the PROM or RAM Character Memory (i.e., the ability to produce symbols for which there is no character associated for dispatching). This mode of operation is designated "transfer mode" and is available only when the Character Generator is selected to "Perform Control Character" mode (PFORM CCHAR, see Section 2.4.3b for details). The manner in which the transfer mode is used to provide these capabilities is detailed for each of the functions below:

a. Load Character Memory

The RAM Character Memory is loaded by outputting a Line Generator Status Command with the PFORM CCHAR bit set, followed by a string of characters of the following form:

<KAaXxXx...Xx>

where:

- < is the Start Transfer character code (SXF, 026).
- K is a lower or upper case K character code (113 or 153).
- Aa are two character codes which are used to designate the beginning address in the RAM Character Memory into which the character generation instructions (XxXx...) are to be loaded. The RAM Character Memory address is formed from Aa in the following manner:

A<6:5:4>=1.

A<3-0>=the 4 most-significant bits of the 10-bit address.

a<6>=1.

a<5-0>=the 6 least-significant bits of the 10-bit address.

For example, to begin loading the RAM from location 0, the following character codes are used:

160,100.

The Character Memory address is incremented after each "Xx" value is loaded, until an End Transfer code is encountered.

Xx are pairs of character codes which specify the value to be loaded into the next location in the RAM Character Memory. The value for each 12-bit memory location is formed from Xx in the following manner:

X<6>=1.

X<5-0>=the 6 most-significant bits of the 12-bit value.

x<6>=1.

x<5-0>=the 6 least-significant bits of the 12-bit value.

For example, to load a "JMS 401" instruction into the RAM, Xx=164,101.

> is the End Transfer character code (EXF,027), or any other character code in the range 0-37.

For example, the sequence of character codes below loads a "JMS 401" instruction into RAM location 2042 and a "RMOV,Dx=5,Dy=0" instruction into location 2043:

26,113,160,142,164,101,101,120,27.

b. Load Coefficient Memory

The RAM portion of the Coefficient Memory is loaded by outputting a Line Generator Status Command with the PFORM CCHAR bit set, followed by a string of characters of the form:

<WapXxXx...Xx>

where:

< is the Start Transfer character code (SXF,026).

W is a lower or upper case W character code (127 or 167).

a is a character code which specifies the Coefficient Memory address where loading is to begin:

A<6>=1.

A<5-0>=Coefficient Memory address in the range
40-77.

The Coefficient Memory address is incremented once for each four "Xx" pairs (typically, once for each set of coefficients A, B, C and D).

p is a character code which specifies which of the coefficients A, B, C or D is to be loaded first. Here, the character to use matches the coefficient name: character A (code 101) for coefficient A, character B for coefficient B, etc. This coefficient pointer "increments" from A through D, then back to A as each "Xx" pair is encountered.

Xx are pairs of character codes which specify the 9-bit values to be loaded for each coefficient.

X<6>=1.

X<5:4:3>=0.

X<2>=speed bit

X<1-0>=sign bits (X<1> must = X<0>).

x<6>=1.

x<5-0>=the 6 least-significant bits of value.

> is the End Transfer character code (EXF,027) or any other character code in the range 0-37.

PS2/MPS Reference Manual
Hardware Details

For example, the following sequence of character codes loads Coefficient Memory address 43 with coefficients A-D for a mid-range character at one-half speed:

26,127,143,101,104,177,100,100,104,100,100,177,27.

c. Load Inter-character and Inter-line Spacing Registers

The Displacement registers referred to in Section 2.4.4.3 may be loaded by outputting a Line Generator Status Command with the PFORM CCHAR bit set, followed by a string of characters of the form:

<PaXxXx...Xx>

where:

- < is the Start Transfer character code (SXF,026).
- P is a lower or upper case P character code (120 or 160).
- a is a character code which specifies which register is to be written first:

a<6>=1.

a<5:4>=0.

a<3-0>=register address in range 10-13 for displacement registers.

The 12-bit integer part, and 12-bit fractional part of X Displacement are written to registers 10 and 12, respectively. The integer and fractional parts of Y Displacement are written to registers 11 and 13.

Xx are pairs of character codes which specify the 12-bit values to be loaded into the Displacement registers.

X<6>=1.

X<5-0>=the 6 most-significant bits of 12-bit value.

x<6>=1.

x<5-0>=the 6 least-significant bits of 12-bit value.

- > is the End Transfer character code (EXF,027) or any other character code in the range 0-37.

PS2/MPS Reference Manual
Hardware Details

The register address is incremented after each "Xx" pair, to load all four values in one transfer, the values must appear in the order: integer x, integer y, fractional x and fractional y. The following sequence of codes loads the X Displacement with 2.6421 (octal) and the Y Displacement with .37 (octal):

26,120,110,100,102,100,101,164,121,137,100,27.

d. Comments Within Character Strings

Comments may be embedded within strings of characters by including the text of the comments in the form:

<"TEXT...">

where:

< is the Start Transfer character code (SXF,026).

" is the quote symbol character code (42).

TEXT... is any string of character codes in the range 040-177 (i.e., all of the 95 displayable ASCII characters).

> is the End Transfer character code (EXF,027) or any other character code in the range 0-37.

If the comment is preceeded by a Line Generator Status command with the PFORM CCHAR bit set, none of the comment characters are displayed; if the PFORM CCHAR bit is not set, the comment characters are displayed.

e. Direct Dispatch to Character Memory Location

Control may be transferred to a specified Character Memory location by outputting a Line Generator Status Command with the PFORM CCHAR bit set, followed by a string of characters of the form:

<DAa>

where:

- < is the Start Transfer character code (SXF,026).
- D is a lower or upper case D character code (104 or 144).
- Aa is a pair of character codes which specifies the starting Character Memory address:
 - A<6>=1.
 - A<5>=0.
 - A<4-0>=the 5 most-significant bits of 11-bit Character Memory address.
 - a<6>=1.
 - a<5-0>=the 6 least-significant bits of 11-bit Character Memory address.
- > is the End Transfer character code (EXF,027) or any other character code in the range 0-37.

Once control has been transferred to location "Aa" in the Character Memory, character generator instructions are executed sequentially (or in the order determined by internal jump instructions) until some form of End is encountered (see Section 2.4.4.4). At this time, the control returns to a mode where the incoming character codes are interpreted for display.

The following sequence of character codes transfers control to Character Memory location 1623:

26,104,116,123,27.

2.4.5 Picture Generator Maintenance Registers

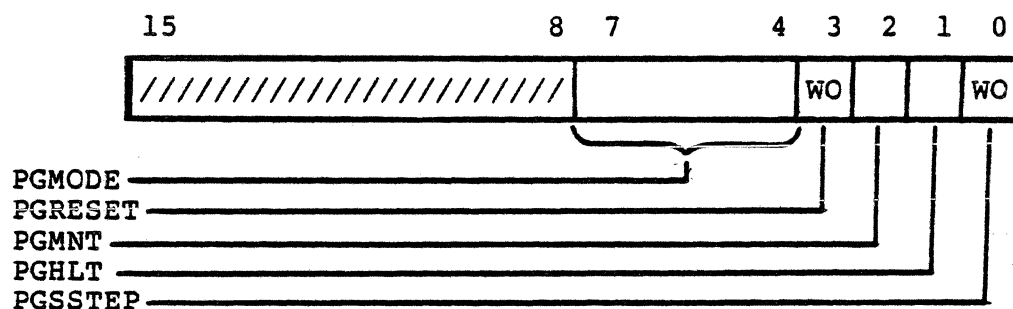
The maintenance registers of the Picture Generator provide access to the internal registers and data paths of the Picture Generator. These registers are provided for maintenance only and need not be accessed during normal operations.

The Picture Generator is controlled by three registers in the SCB, shown in Figure 2.4-13. These registers are described in detail in the following sections.

SCB: 177740	PGSR
177741	PGXBUS
177742	PGYBUS
177743	not used

Figure 2.4-13
Picture Generator Maintenance Registers

a. Picture Generator Status Register (PGSR)-SCB:177740



The PGSR is used to provide maintenance control of the Picture Generator.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
7-4	PGMODE	PGMODE is used to select which internal Picture Generator registers can be read via the PGXBUS and PGYBUS registers. (See below). Cleared by RESET.
3	PGRESET	When this bit is set, it causes a RESET signal to be issued to the Line Generator, Character Generator and Refresh Controller. These will be reset to their initial power-up state. Write only. Always reads as a zero.
2	PGMNT	Setting PGMNT, enables the Picture Generator internal registers to be read by reading the PGXBUS and PGYBUS registers. The PGMODE field is used to select which registers are to be accessed. PGMNT must be set for the registers to be accessed. Cleared by RESET.
1	PGHLT	When PGHLT is set, it disables the Picture Generator system clock which normally issues a pulse each 100 nsec. This freezes the Picture Generator for diagnostic purposes. Cleared by RESET.

0 PGSSTEP This bit is set to issue a single clock pulse to the Picture Generator. Write only. Always reads as a zero.

b. Picture Generator X-Bus (PGXBUS)-SCB:177741

This register is used to read the data on the main internal X-Bus of the Picture Generator. When the Picture Generator is not in maintenance mode (PGMNT=0), this register will show the value of whichever register currently has control of the X-Bus. Note that the Line Generator Input FIFO will always have control of this bus unless a Character Generator command is being processed. When the Picture Generator is halted in maintenance mode (i.e., PGHLT=1 and PGMNT=1), the internal registers connected to the X-Bus may be read by setting the appropriate PGMODE bits of the PGSR. Read only.

The following table details the data that may be read from this register for each of the PGMODE values. If the content of a bit field is not specified, then it should be considered undefined.

Line Generator Maintenance Paths

PGMODE	Set PGSR	Read PGXBUS	Values Obtained
0	6	<13-9>	Output control ROM current address <4-0>.
1	26	<15-11>	Input control ROM current address <4-0>.
2	46	<15-12>	X Normalize code <3-0>.
		<11-0>	X ALU B input <11-0>.
3	66	<15>	X ALU F output <12> (delayed 1 clock).
		<14>	X ALU inverted carryout (delayed 1 clock).
		<13>	X ALU inverted carryout.
		<12-0>	X ALU F output <12-0>.
4	106	<11-8>	Sum of squares <8-5>.
		<7-0>	(Normalized delta y) squared <7-0>.
5	126	<11-8>	sum of squares <4-1> (bit 0 discarded).
		<7-0>	(Normalized delta x) squared <7-0>.
6	146	<11-8>	Denormalized square root <15-12>.
		<7-0>	Reciprocal of square root <7-0>.
7	166	<11-0>	Denormalized square root <11-0>.
10-11			Unused.

Character Generator Maintenance Paths

<u>PGMODE</u>	<u>Set</u> <u>PGSR</u>	<u>Read</u> <u>PGXBUS</u>	<u>Values Obtained</u>
12	246	<7-6> <5-0>	Style field of Font Stack. Coefficient address of Font Stack (Coefficient Address + Coefficient Offset).
13	266	<13-12> <11-0>	Character intensity <3-2>. X-character multiplier output <11-0>.
14	306	<11-0>	Value register.
15	326	<10-0>	SAVE CHADR register (only if EXT=0).
16	346	<11-0>	Character Memory output (only if EXT=0).
17	366	<7-0>	Character Generator control ROM current ad- dress.

c. Picture Generator Y-Bus (PGYBUS)-SCB:177742

This register, like the PGXBUS, is used to read the data on the main internal Y-Bus of the Picture Generator. When the Picture Generator is not in maintenance mode (PGMNT=0), this register will contain the value of whichever register currently has control of the Y-Bus. Note again that the Line Generator Input FIFO will always have control of the bus unless a Character Generator command is being processed. When the Picture Generator is halted in maintenance mode (PGHLT=1, PGMNT=1), the internal registers connected to the Y-Bus may be read by setting the appropriate bits of the PGSR. Read only.

The following table details the data that may be read from this register for each of the PGMODE values. If the content of a bit field is not specified, it should be considered undefined.

Line Generator Maintenance Paths

<u>PGMODE</u>	<u>Set PGSR</u>	<u>Read PGYBUS</u>	<u>Values obtained</u>
0	6	<4-0>	Arithmetic control ROM current address <4-0>.
1	26	<4-0>	Multiplier control ROM current address <4-0>.
2	46	<15-12>	Y normalize code <3-0>.
		<11-0>	Y ALU B input <11-0>.
3	66	<15>	Y ALU F output <12> (delayed 1 clock).
		<14>	Y ALU inverted carryout (delayed 1 clock).
		<13>	Y ALU inverted carryout.
		<12-0>	Y ALU F output <12-0>.
4	106	<11>	Inverted X D/A input <11>.
		<10-0>	X D/A input <10-0>.
5	126	<11>	Inverted Y D/A input <11>.
		<10-0>	Y D/A input <10-0>.
6	146	<7-2>	Z value.
		<1-0>	Unshifted delta Z <4-3>.
7	166	<7>	Unshifted delta Z sign.
		<6-0>	Unshifted delta Z <11-5>.
10-12			Unused.

Character Generator Maintenance Paths

<u>PGMODE</u>	<u>Set PGSR</u>	<u>Read PGYBUS</u>	<u>Values Obtained</u>
13	266	<15-14>	Character intensity <1-0>.
		<11-0>	Y-character multiplier output.
14-17			Unused.

The Character and Coefficient Memories of the Character Generator may be read using the maintenance registers by executing special transfer functions similar to those described in Section 2.4.4.5.

The procedure for reading the Character Memory is as follows:

1. Execute the Read Character Memory transfer function (see below), which specifies the first address to be read.
2. Place the Picture Generator into single-step mode by setting the PGHLT bit in the PGSR register.
3. Place the Picture Generator into maintenance mode by setting the PGMNT bit in the PGSR register.
4. Set the PGMODE bits to 16 in the PGSR register. This drives the output from the Character Memory to the Character Generator XCBUS.
5. Single-step the Picture Generator by setting the PGSSTEP bit in the PGSR register.
6. Read the Character Memory output from the PGXBUS register.
7. Repeat steps 5 and 6 to read successive locations in the Character Memory.
8. Return the Picture Generator to normal "run-free" mode by clearing the PGMNT, then the PGHLT bit in the PGSR register.

The Read Character Memory transfer function is executed by outputting a Line Generator Status command with the PFORM CCHAR bit set, followed by a string of characters of the form:

<MAa>

where:

- < is the Start Transfer control code (SXF,026).
- M is a lower or upper case M character code (115 or 155).
- Aa is a pair of character codes which specifies the Character Memory address at which reading is to begin.
 - A<6>=1.
 - A<5>=0.
 - A<4-0>=the 5 most-significant bits of the 11-bit Character Memory address.
 - a<6>=1.
 - a<5-0>=the 6 least-significant bits of the 11-bit Character Memory address.
- > is the End Transfer control code (EXF,027).

The procedure for reading the Coefficient Memories is as follows:

1. Execute a "Push-Load Font Stack" function with arbitrary data. This saves the current font data. Example: bytes 3, 74, 0, 0.
2. Execute the Read Coefficient Memory transfer function (see below) with the parameters set to read either the A and C coefficients, or the B and D coefficients.
3. Place the Picture Generator into single-step and maintenance mode by setting the PGHLT and PGMNT bits in the PGSR register. Also, set PGMODE=12 (e.g., set PGSR=247). This puts the Coefficient Memory address on the XCBUS.
4. Repeat step 3.
5. Repeat step 3. This loads the font stack with the desired address and produces the contents of the Coefficient RAM (A, B, C, or D) on the output of the Y-character multiplier.
6. Set the PGSR to 266 (PGHLT, PGMNT and PGMODE=13). This reads the Y-character multiplier and puts the 7 bits of data sign extended into bits <11-0> of the PGYBUS register.
7. Repeat steps 5 and 6 to read successive Coefficient Memory data. The order of appearance is described below.
8. Return the Picture Generator to "free run" mode but leaving PGMNT bit set in the PGSR (PGSR=4).
9. Return the Picture Generator to "free-run" operation by clearing the PGSR (PGSR=0).
10. Execute a "Pop Font Stack" function to restore the Font Parameter Stack to its initial state.

The Read Coefficient Memory transfer function is executed by outputting a Line Generator Status Command with the PFORM CCHAR bit set, followed by a string of characters of the form:

<]apMm>

where:

- < is the Start Transfer character code (SXF,026).
-] is the close bracket character code (135).
- a is the character code which specifies the address in the Coefficient Memory at which reading is to begin:
 - a<6>=1.
 - a<5-0>=Coefficient Memory address.
 - 0-17=Coefficient ROM.
 - 20-37=nothing.
 - 40-77=Coefficient RAM
- p is a character code which specifies which Coefficient (A, B, C or D) will be read first. The character here is taken literally; i.e., to begin reading at coefficient A, the character code for upper case A is used (code 101).
- Mm is a pair of character codes which specify whether the character multiplier will produce the A and C coefficients on the output, or the B and D coefficients. To get the A and C coefficients set M=100, m=120. To get the B and D coefficients, set M=100, m=101.

When reading the data from the Coefficient Memory, only the A and C, or B and D coefficients may be read on a given execution of the above procedure. However, the Picture Generator will still produce data in cycles of 4: either C, A, A, C or D, B, B, D. The following examples illustrate this. The "/" used indicates the start of this cycle. For the starting location (n) specified, data are produced in the order:

<u>Start On</u>	<u>Data returned on successive reads</u>
A(n)	A(n),A(n),C(n)/C(n+1),A(n+1),A(n+1),C(n+1)/C(n+2),...
B(n)	B(n),D(n)/D(n+1),B(n+1),B(n+1),D(n+1)/D(n+2),...
C(n)	C(n)/C(n+1),A(n+1),A(n+1),C(n+1)/C(n+2),...
D(n)	/D(n+1),B(n+1),B(n+1),D(n+1)/D(n+2),...

PS2/MPS Reference Manual
Hardware Details

Note that to start reading on coefficient D, a starting address of one lower than desired should be specified (e.g., to start reading at D(40), specify D(37)).

2.5 Interrupt Control

As described in Section 2.2, the Picture System/PDP-11 Interface is the only Picture System device interfaced directly to the UNIBUS of the PDP-11. As such, the DMA channel is under direct interrupt control of the PDP-11 with READY, Interrupt Enable (DMAIE) and Extended Bus Address (XBA) bits available in the IOST register as with all standard DEC NPR device controllers. However, there are many Picture System devices available for use under interrupt control from the PDP-11 other than the I/O interface. These devices are interfaced to the Picture System PSBUS and not directly to the UNIBUS of the PDP-11. Three PDP-11 interrupt vectors are available to initiate the appropriate interrupt handler routine for these Picture System interrupt processes. These three interrupt vectors, in addition to the DMA interrupt vector, are assigned the following priority levels (in descending order) and standard interrupt vectors as follows:

- | | | |
|----|------------------------------------|-----------|
| 1. | Real-Time Clock Interrupt Control: | BR5 - 340 |
| 2. | System Interrupt Control: | BR5 - 344 |
| 3. | Device Interrupt Control: | BR5 - 350 |
| 4. | DMA Interrupt Control: | BR5 - 354 |

The DMA Interrupt Control is described in detail in Section 2.2. Control of the Real-Time Clock, System and Device interrupt processes is supervised by a Picture System Interrupt Enable bit (PSIE) in the IOST register and individual interrupt enable bits for each of the various interrupt processes. The Picture System 2 utilizes all of these interrupt processes in the processing and display of pictures. The Multi Picture System, however, relies on the Refresh and Device Processor for line frequency synchronization, refresh and device monitoring, and hence utilizes only the DMA and System interrupt processes. The Real-Time Clock, System and Device Interrupt Control are described for the Picture System 2 in Section 2.5.1 below. The System Interrupt Control for the Multi Picture System is detailed in Section 2.5.2.

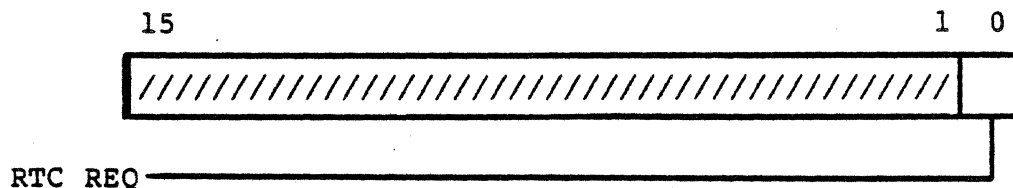
2.5.1 Picture System 2 Interrupt Control

The Picture System 2 utilizes the Real-Time Clock, System and Device interrupt processes in addition to the DMA interrupt process detailed previously. The Picture Controller must provide an interrupt service routine for each of these interrupt processes that are to be used. The enabling of each of the independent interrupt processes is supervised by the individual interrupt enable bits detailed below.

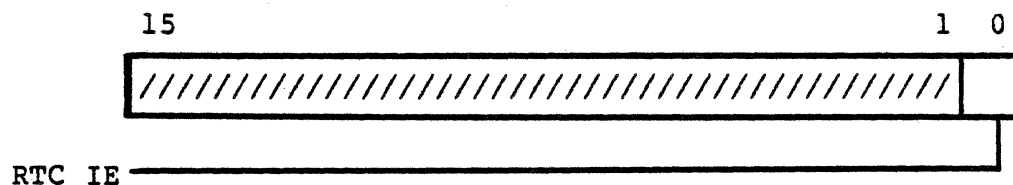
2.5.1.1 PS2 Real-Time Clock Interrupt Control

The Real-Time Clock is used to control refreshing of the Picture System at line frequency intervals under interrupt control. The Real-Time Clock has two registers in the SCB assigned to regulate this interrupt--the Real-Time Clock Interrupt Request and the Real-Time Clock Interrupt Enable registers.

a. Real-Time Clock Interrupt Request (RTCREQ)-SCB:177760



b. Real-Time Clock Interrupt Enable (RTCIE)-SCB:177761



The Real-Time Clock Interrupt Request and Real-Time Clock Interrupt Enable registers have a corresponding bit assigned in each to control the interrupt process for the Real-Time Clock. The RTC IE bit is set in the RTCIE register to allow an interrupt to be issued to the host computer whenever the RTC REQ bit becomes set in the RTCREQ register. Both the RTCREQ and RTCIE registers are cleared by RESET. The conditions which cause a request to be issued are detailed below.

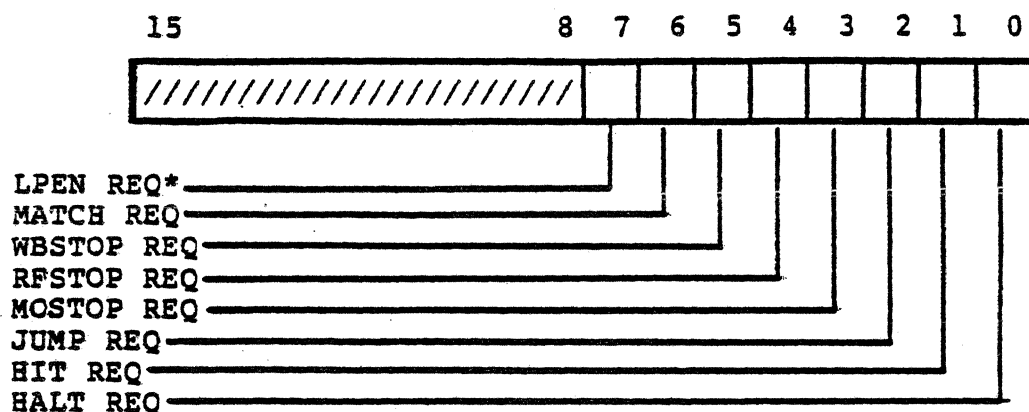
PS2/MPS Reference Manual
Hardware Details

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-1	not used	
0	RTC REQ	RTC REQ is set by the Real-Time Clock controller whenever the line frequency counter 1 (CNT1) has elapsed. If counter 1 (CNT1) is selected for external syncing to a source other than the line frequency, RTC REQ is set when the count elapses. (See Section 2.6.1 for specific details.) RTC REQ must be cleared to acknowledge a RTC interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot programmably be set to a 1.

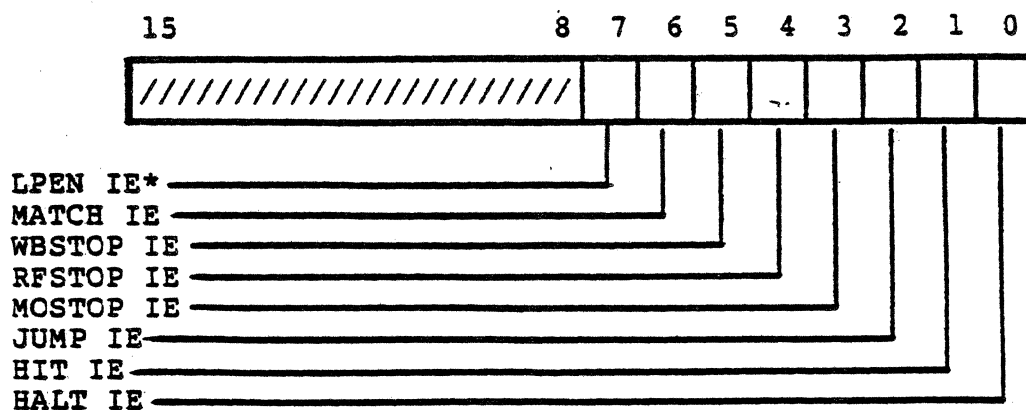
2.5.1.2 PS2 System Interrupt Control

There are several Picture System functions which have been designated System Control functions and have the ability to cause the processing of data in the Picture System to cease and an interrupt to be issued to the host computer. These System Control functions have two registers in the SCB assigned to regulate these interrupt processes--the System Interrupt Request and System Interrupt Enable registers.

a. System Interrupt Request (SYSREQ)-SCB:177762



b. System Interrupt Enable (SYSIE)-SCB:177763



The System Interrupt Request and System Interrupt Enable registers have corresponding bits assigned for each System Control function. A bit in the System Interrupt Enable register is set to allow an interrupt to be issued to the host computer whenever the corresponding REQ bit becomes set in the System Interrupt Request register. The bits in the System Interrupt Request re-

gister are set by the Picture Processor or Refresh Controller to indicate the presence of a condition which may require the attention of the control program. Both the SYSREQ and SYSIE registers are cleared by RESET. The conditions which cause a request to be issued are detailed for each of the REQ bits below.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-8	not used	
7	LPEN REQ*	LPEN REQ is set by the Light Pen controller whenever a light pen sensitive line, dot or character is detected by the photosensor of the light pen or a light pen tip switch transition occurs. LPEN REQ must be acknowledged to allow any further light pen hits or tip switch transitions to be detected. LPEN REQ must be cleared to acknowledge a LPEN interrupt. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
6	MATCH REQ	MATCH REQ is set by the Refresh Controller whenever the RFCSN=RFSN and indicates that a segment name match has been encountered. Note that the registers need not be exactly equal depending on the current status of the SEARCH MODE bits of the RFSR. If the MATCH HOLD bit in the RFSR is set when the MATCH REQ is set, the MATCH REQ bit must be cleared before the current refresh cycle will continue. MATCH REQ must be cleared to acknowledge a MATCH interrupt. Cleared by RESET or loading

*For Light Pen interface (PS card #195273-100) of ECO level A1 or higher, the LPEN REQ bit has been moved to bit 7 of the Device Interrupt Request 16-31 register (DEVREQ 16-31) and the LPEN IE bit has been moved to bit 7 of the Device Interrupt Enable 16-31 register (DEVIE 16-31). See Section 2.5.1.3.

- with a 1. Cannot be programmably set to a 1.
- 5 WBSTOP REQ WBSTOP REQ (Writeback Stopped) is set by the Refresh Controller to indicate that the RFAWA has reached the limit in PSMEM as determined by the RFAWL, during writeback processing. Writeback may be continued by resetting the RFAWA or the RFAWL and clearing the WBSTOP REQ bit. Writeback may be stopped by clearing the WRITEBACK bit of the RFSR. WBSTOP REQ must be cleared to acknowledge a WBSTOP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
- 4 RFSTOP REQ RFSTOP REQ (Refresh Stopped) is set by the Refresh Controller whenever the RFAIA=RFAIL or a Refresh Control command with the HALT bit set is encountered by the Refresh Controller. RFSTOP REQ must be cleared to acknowledge a RFSTOP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
- 3 MOSTOP REQ MOSTOP REQ (MAP Output Stopped) is set by the Picture Processor to indicate that all output from the MAP Output Formatter has ceased because the MAOA has reached the limit in PSMEM as determined by the MAOL. Output may be continued by resetting the MAOA or MAOL and clearing the MOSTOP REQ bit. MOSTOP REQ must be cleared to acknowledge a MOSTOP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.

2 JUMP REQ

JUMP REQ is set by the Picture Processor whenever the MAP Input Controller is functioning as a Passive device and the MAP has just executed a JUMP, PUSHJ, or POPJ RSR command. The Picture Processor is in the HOLD state while JUMP REQ is set. When servicing the JUMP interrupt, the control program in the host computer can clear the MAP input FIFO by resetting the MAIA register and then start a new DMA transfer from the host computer memory corresponding to the location specified for the JUMP, PUSHJ or POPJ command. JUMP REQ must be cleared to acknowledge a JUMP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.

1 HIT REQ

HIT REQ is set by the Picture processor whenever a point or line which is being processed by the MAP is not entirely clipped from the scene. This indicates that a point or line is within or passes through the current window. HIT REQ must be cleared to acknowledge a HIT interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to 1.

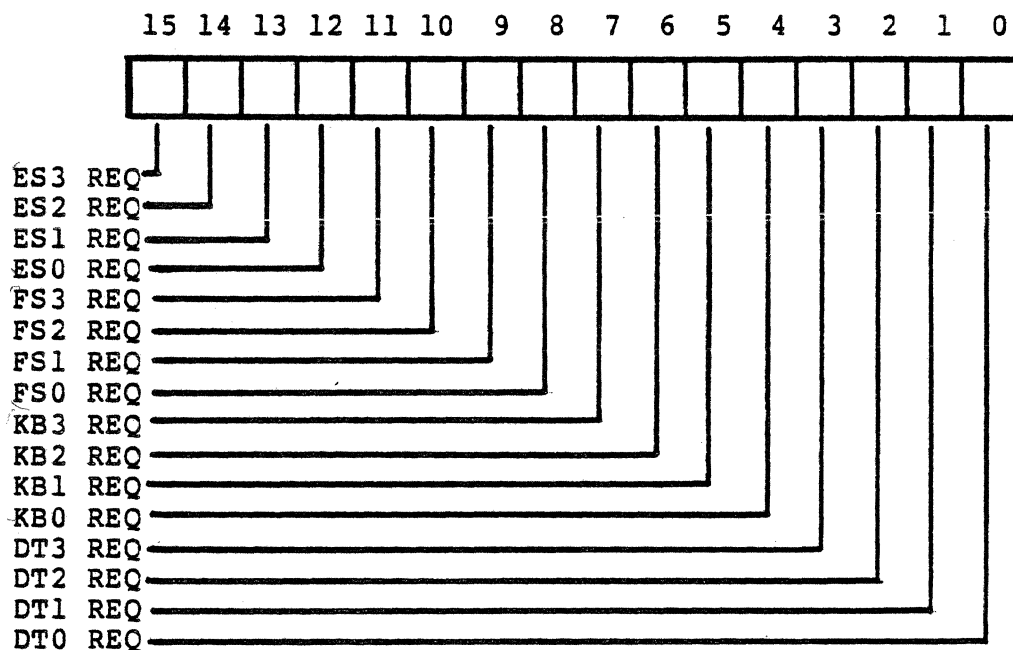
0 HALT REQ

HALT REQ is set by the Picture Processor whenever the MAP has just executed a HALT RSR command. The Picture Processor is in the HOLD state while HALT REQ is set. HALT REQ must be cleared to acknowledge a HALT interrupt and/or to restart the MAP after a HALT RSR command has been executed. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.

2.5.1.3 PS2 Device Interrupt Control

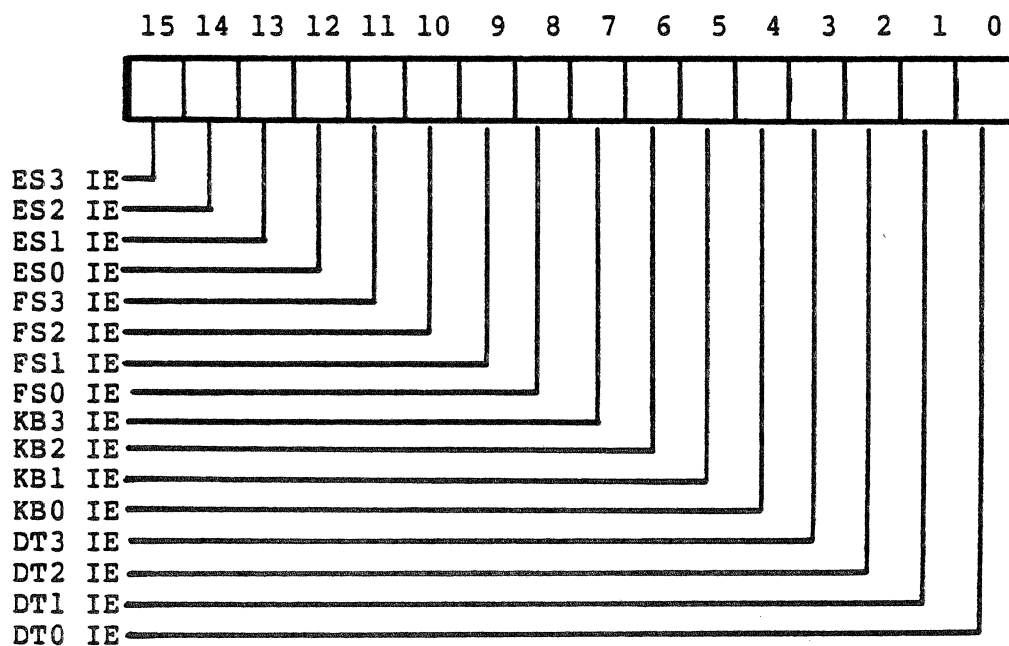
All other Picture System devices (i.e., all those but the Real-Time Clock, Picture Processor and Refresh Controller) which may be used under interrupt control are assigned a set of Interrupt Request and Enable bits in the Device Interrupt Control registers. There are four registers in the SCB assigned to regulate device interrupts--two Device Interrupt Request and two Device Interrupt Enable registers.

a. Device Interrupt Request 0-15 (DEVREQ 0-15)-SCB:177764



PS2/MPS Reference Manual
Hardware Details

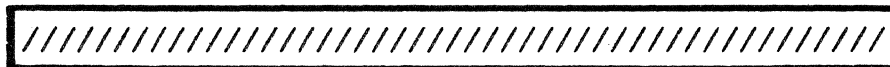
b. Device Interrupt Enable 0-15 (DEVIE 0-15)-SCB:177765



- c. Device Interrupt Request 16-31 (DEVREQ 16-31)-SCB:177766

15

0

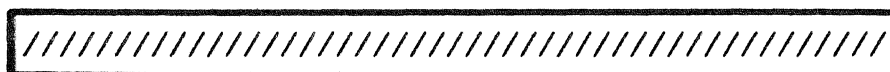


Unassigned*

- d. Device Interrupt Enable 16-31 (DEVIE 16-31)-SCB:177767

15

0



Unassigned*

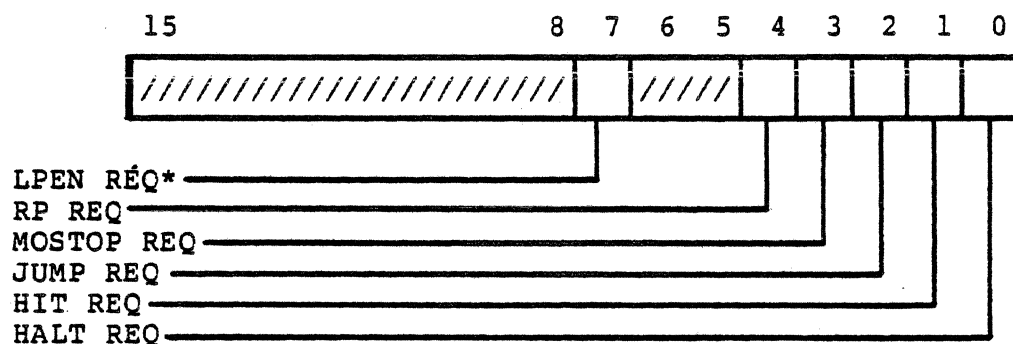
The Device Interrupt Request and Device Interrupt Enable registers have corresponding bits assigned for each Picture System device which may be used under interrupt control. A bit in a Device Interrupt Enable register is set to allow an interrupt to be issued to the host computer whenever the corresponding REQ bit becomes set in the Device Interrupt Request register. The REQ bits are set by the individual device controllers to indicate the presense of a condition which may require the attention of the control program. The conditions which cause a request to be issued are detailed for each of the devices in Section 2.6. The devices for which interrupt request and enable bits are currently assigned are the Data Tablet (DT), the Alphabetic Keyboard (KB), the Function Switches (FS) and the Extended Switches (ES). Both sets of DEVREQ and DEVIE registers are cleared by RESET.

*For Light Pen interface (PS card #195273-100) of ECO level A1 or higher, the LPEN REQ bit has been moved to bit 7 of the Device Interrupt Request 16-31 register (DEVREQ 16-31) and the LPEN IE bit has been moved to bit 7 of the Device Interrupt Enable 16-31 register (DEVIE 16-31). See Section 2.5.1.3.

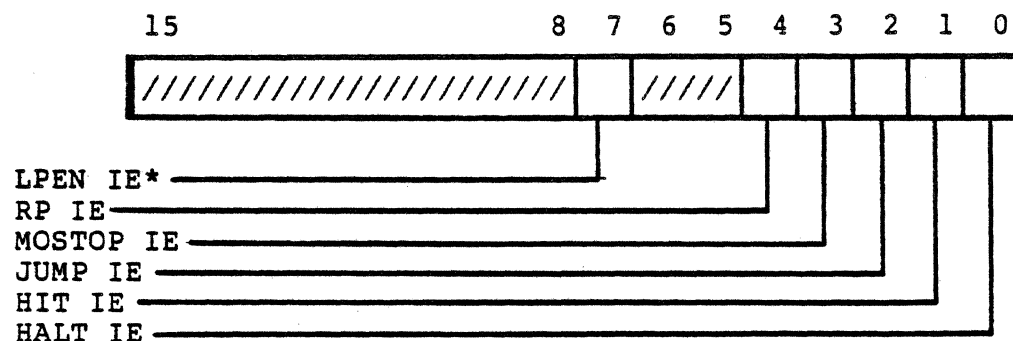
2.5.2 Multi Picture System Interrupt Control

The Multi Picture System utilizes only the System interrupt process in addition to the DMA interrupt process detailed previously. As described in Section 2.4.2.2, the Refresh and Device Processor supervises the refreshing and device monitoring previously required of the Picture Controller by the Picture System 2. However, there are several Multi Picture System functions which have been designated System Control functions and have the ability to cause an interrupt to be issued to the host computer. These System Control functions have two registers in the SCB assigned to regulate these interrupt processes--the System Interrupt Request and System Interrupt Enable registers.

a. System Interrupt Request (SYSREQ)-SCB:177762



b. System Interrupt Enable (SYSIE)-SCB:177763



The System Interrupt Request and System Interrupt Enable registers have corresponding bits assigned for each System Control function. A bit in the System Interrupt Enable register is set to allow an interrupt to be issued to the host computer whenever the corresponding REQ bit becomes set in the System Interrupt Request

register. The bits in the System Interrupt Request register are set by the Picture Processor or Refresh Controller to indicate the presence of a condition which may require the attention of the control program. Both the SYSREQ and SYSIE registers are initialized by RESET. The conditions which cause a request to be issued are detailed for each of the REQ bits below.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-8	not used	
7	LPEN REQ*	LPEN REQ is set by the Light Pen controller whenever a light pen sensitive line, dot or character is detected by the photosensor of the light pen or a light pen tip switch transition occurs. LPEN REQ must be acknowledged to allow any further light pen hits or tip switch transitions to be detected. LPEN REQ must be cleared to acknowledge a LPEN interrupt. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
6-5	not used	
4	RP REQ	RP REQ (Refresh Processor) is set when the Refresh and Device Processor has encountered a condition that requires service by the host computer. The setting of the RP REQ bit indicates that a bit has been set in the Interrupt Bits 31-0 of the Refresh Table that resides in Picture Memory. The Interrupt Bits 31-0 must be interrogated to determine the corresponding Command Address in the Refresh Table that

*For Light Pen interface (PS card #195273-100) of ECO level A1 or higher, the LPEN REQ bit has been moved to bit 7 of the Device Interrupt Request 16-31 register (DEVREQ 16-31) and the LPEN IE bit has been moved to bit 7 of the Device Interrupt Enable 16-31 register (DEVIE 16-31). See Section 2.5.1.3.

- caused the request. RP REQ must be cleared to acknowledge an RP interrupt. Read/Write. Set by RESET. Cleared by loading with a 1. Cannot be programmably set to a 1.
- 3 MOSTOP REQ MOSTOP REQ (MAP Output Stopped) is set by the Picture Processor to indicate that all output from the MAP Output Formatter has ceased because the MAOA has reached the limit in PSMEM as determined by the MAOL. Output may be continued by resetting the MAOA or MAOL and clearing the MOSTOP REQ bit. MOSTOP REQ must be cleared to acknowledge a MOSTOP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
- 2 JUMP REQ JUMP REQ is set by the Picture Processor whenever the MAP Input Controller is functioning as a Passive device and the MAP has just executed a JUMP, PUSHJ, or POPJ RSR command. The Picture Processor is in the HOLD state while JUMP REQ is set. When servicing the JUMP interrupt, the control program in the host computer can clear the MAP input FIFO by resetting the MAIA register and then start a new DMA transfer from the host computer memory corresponding to the location specified for the JUMP, PUSHJ or POPJ command. JUMP REQ must be cleared to acknowledge a JUMP interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.
- 1 HIT REQ HIT REQ is set by the Picture processor whenever a point or line which is being processed by the MAP is not entirely

clipped from the scene. This indicates that a point or line is within or passes through the current window. HIT REQ must be cleared to acknowledge a HIT interrupt. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to 1.

0

HALT REQ

HALT REQ is set by the Picture Processor whenever the MAP has just executed a HALT RSR command. The Picture Processor is in the HOLD state while HALT REQ is set. HALT REQ must be cleared to acknowledge a HALT interrupt and/or to restart the MAP after a HALT RSR command has been executed. Read/Write. Cleared by RESET or loading with a 1. Cannot be programmably set to a 1.

2.6 Picture System Devices

All devices which interface to Picture System are controlled by a set of device registers in the System Control Block. These device registers may be read or written under program control from the host computer by using the Direct I/O interface to the Picture System. There are seven standard device interfaces available for use with the Picture System:

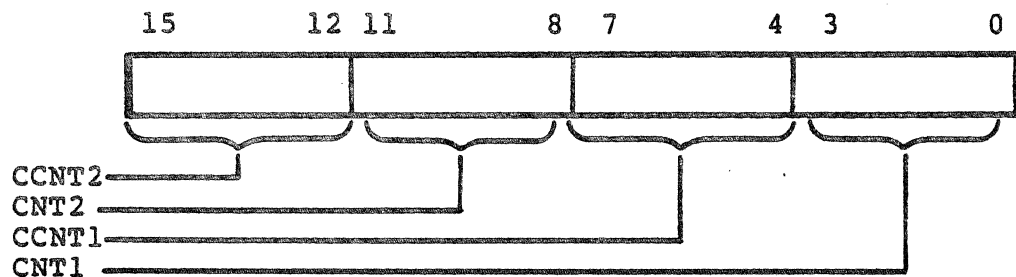
1. Real-Time Clock
2. Data Tablet
3. Alphanumeric Keyboard
4. Function Switches & Lights/Lighted Function Buttons
5. Control Dials/Joystick
6. Light Pen
7. Remote Terminal Interface

Detailed descriptions for each of these devices and their associated interfaces are contained in the following sections.

2.6.1 Real-Time Clock

For the Picture System 2, the Real-Time Clock is used to provide a source of timing to the host computer allowing synchronization of the display of data with the frequency of the power source (i.e., 60 or 50 Hz.), or with an external source such as a camera synchronization signal. This device is not available for the Multi Picture System (see Section 2.4.2.2e and 2.4.2.2f for the functional equivalent device of the Multi Picture System). The Real-Time Clock actually provides two clocks (CLOCK1 and CLOCK2) for which twice the line frequency or a user-supplied external input can be selected as the basic timing. Associated with each clock is a 4-bit counter (CNT1 and CNT2) which is used to divide the basic timing providing the final clock frequency. CLOCK1 is used to interrupt the host computer while CLOCK2 is used to provide timing for the automatic refresh feature of the Refresh Controller (see Section 2.4.2 for automatic refresh details). The clocks are controlled by four registers--RTCCNT, RTCSR, RTCREQ and RTCIE. RTCREQ and RTCIE are located in the SCB at 177760 and 177761 as described in Section 2.5.1. The RTCCNT and RTCSR are described in detail below:

a. Real-Time Clock Counts (RTCCNT)-SCB:177744

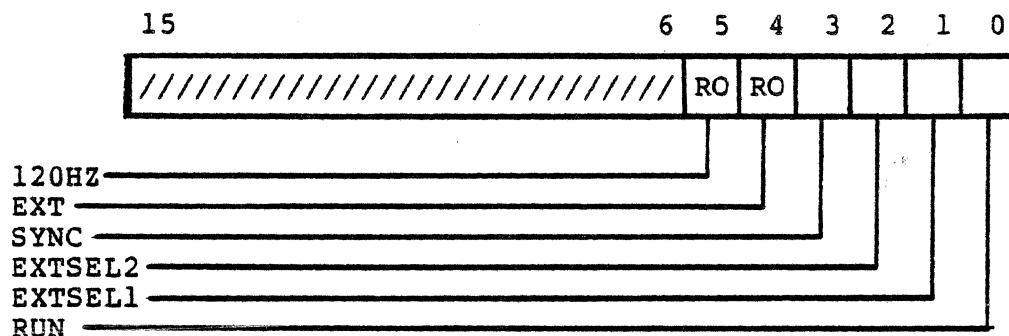


The RTCCNT register is used to select the final clock frequencies by selecting the increments by which the basic timing is to be divided. The division is accomplished by incrementing the CCNT values (CCNT2 and CCNT1) at each clock pulse until the count is equal to all ones. The current value of the clock(s) may be read at any time by reading the current count fields (CCNT2 and CCNT1). When either CCNT2 or CCNT1 equals all ones, the field is automatically reloaded with the value in the associated CNT field allowing the clock to continue to run at the same rate without program supervision. When CCNT1 goes to all ones, it sets the RTCREQ bit in the RTCREQ register, issuing an interrupt to the host computer if the RTCIE bit is enabled. When

CCNT2 goes to all ones, it initiates an automatic refresh cycle by the Refresh Controller if the AUTOREF bit is set in the RFSR (see Section 2.4.2 for automatic refresh details).

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-12	CCNT2	CCNT2 is a read-only field that reflects the current count of CLOCK2. Incremented by one for each clock pulse and set to the value in CNT2 when CCNT2 equals all ones.
11-8	CNT2	CNT2 defines the value the basic timing of CLOCK2 is to be divided by. Read/Write. Set to 1110 (60 Hz.) by RESET.
7-4	CCNT1	CCNT1 is a read-only field that reflects the current count of CLOCK1. Incremented by one for each clock pulse and set to the value in CNT1 when CCNT1 equals all ones.
3-0	CNT1	CNT1 defines the value by which the basic timing of CLOCK1 is to be divided. Read/Write. Set to 1111 (120 Hz.) by RESET.

b. Real-Time Clock Status Register (RTCSR)-SCB:177745



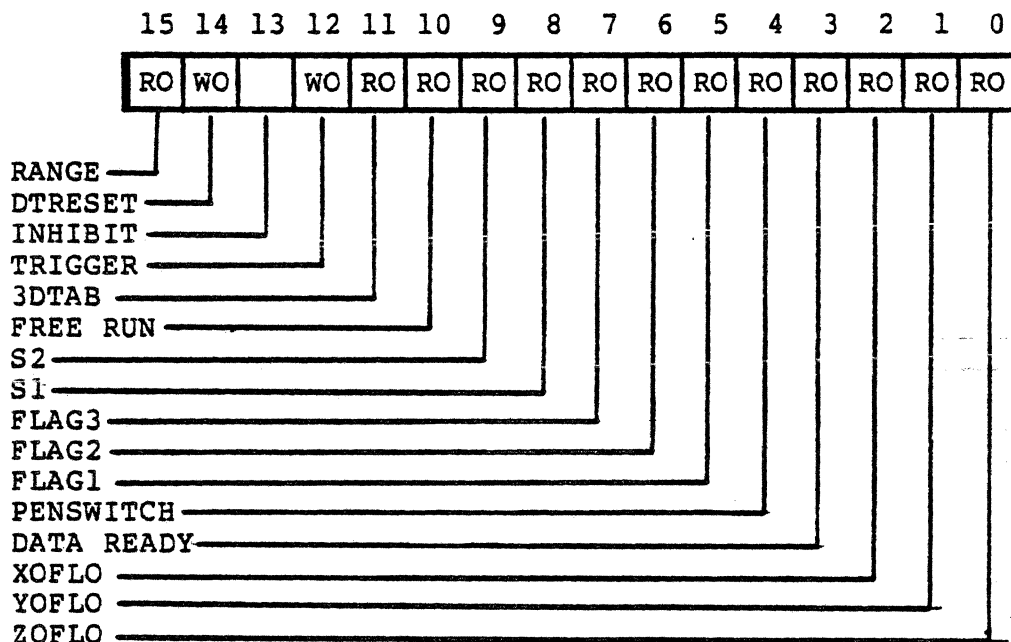
PS2/MPS Reference Manual
Hardware Details

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-6	not used	
5	120HZ	The basic line frequency can be determined from this bit. This bit is driven by a signal which is twice the line frequency. The duty cycle of the signal is such that the bit is clear for approximately 10% of the cycle. This bit stays clear for a short duration so that it will usually be read as a one. Used for diagnostic purposes. Read only.
4	EXT	The basic external timing source can be read by this bit. This is the basic Picture System one-bit interface for such things as movie camera interfaces. Read only.
3	SYNC	SYNC causes automatic refresh timing to be taken from CLOCK1 so that interrupt and automatic refresh timing can be synchronized as well as run at the same rate. Read/Write. Cleared by RESET.
2	EXTSEL2	Selects external basic timing for CLOCK2. Read/Write. Cleared by RESET.
1	EXTSEL1	Selects external basic timing for CLOCK1. Read/Write. Cleared by RESET.
0	RUN	RUN is set to allow the individual clock pulses from CLOCK1 to increment the CLOCK1 counter CCNT1. Read/Write. Cleared by RESET.

2.6.2 Data Tablet

The Data Tablet is controlled by four data registers and two interrupt control registers in the SCB. The Interrupt Request and Interrupt Enable registers are described in Section 2.5.3. The data registers are the Data Tablet Control and Status Register and the X, Y and Z Data registers detailed below:

a. Data Tablet Status Register (DTSR)-SCB:177664



The Data Tablet interface has been flexibly designed so that interfacing with many different types of tablets (i.e., magnetic, electrostatic, two- or three-dimensional, etc.) is possible. Picture System typically is supplied with a Summagraphics Data Tablet. For this tablet, up to five milliseconds are required to convert pen positional information to coordinate data. This conversion can be initiated periodically by the tablet controller (stream mode) or by the user's program (remote mode). Failure of a conversion is a result of the setting of any overflow bit, indicating that the pen position relative to that axis could not be determined. A switch selectable option on the interface determines under what conditions the tablet can be made to request an interrupt (i.e., to set the Data Tablet Request bit, DT REQ, in the Device Interrupt Request register). The DTREQ bit is normally set at the

successful or unsuccessful completion of a tablet conversion. Optionally, the tablet interface is DIP switch-selectable to allow the setting of the DTREQ bit only at the successful completion of a tablet conversion.

<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	RANGE*	Reads 1 if the pen is out of range of the tablet surface, 0 if the pen is within range of the surface (nominally approximately 3/8 inch from the tablet surface).
14	DTRESET	Reset the tablet interface and controller. Write only. Always reads as a zero.
13	INHIBIT	Set to suspend initiation of tablet data conversions from any source (remote trigger or stream mode).
12	TRIGGER	Initiate a data tablet controller conversion (used with tablet in remote mode). Write only. Always reads as a zero.
11	3DTAB	Dip switch settable status bit used to denote that the tablet interface is to be used for a 3D tablet. This enables the setting of the ZOFLO bit and causes Z DATA value to be updated in the Data Tablet Z Data (DTZDAT) register. Read Only.
10	FREE RUN**	Dip switch settable status bit used to denote that the tablet interface is to be used in free run mode. When this bit is set, it indicates that the tablet controller is enabled to start another data conversion immediately after the previous

*This function is available only on tablet interfaces (PS card #195181-100) of ECO level A2 or higher.

**This function is available only on tablet interfaces (PS card #195181-100) of ECO level A1 or higher.

one has completed. If the bit is clear, it means that the DTREQ bit must be acknowledged before another data conversion will begin. The FREE RUN switch must be set for the Multi Picture System and is normally not set for the Picture System 2. Read only.

9-8	S2-S1	Dip switch settable status bits used to denote individual tablet characteristics. Read only.
7-5	FLAG3,2,1	Status bits associated with tablet options (cursor with multiple switches, etc.). Read only.
4	PENSWITCH	Reads 1 if the pen switch is closed (pen is down), 0 if open (pen is up). Read only.
3	DATA READY	Set once a successful conversion is complete (not out of range and no overflow). Causes data values to be updated and if FREE RUN=0 (see above) inhibits further data conversions until the DTREQ bit is acknowledged. This assures the values in the data registers are a result of the same conversion. Cleared by acknowledging the DTREQ bit or RESET.
2	XOFLO*	Set if the pen was out of range in respect to the X-axis when a data conversion was completed. Read only. Cleared by initiation of a new conversion or RESET.

*For the Summagraphics Series ID Data Tablet, the XOFLO and YOFLO bits are actually sign bits indicating the pen position with respect to the origin. The origin is operator selectable and restores to the lower left corner on powerup or RESET. This is the typical origin setting for use with the Picture System. XOFLO and YOFLO will always be 0 in this case.

Note: Completion of a data conversion attempt results in setting the DATA READY bit (3) or one of the X, Y or Z overflow bits (2-0).

1	YOFLO	Set if the pen was out of range with respect to the Y-axis when a data conversion was completed. Read only. Cleared by initiation of a new conversion or RESET.
0	ZOFLO	Set if the pen was out of range with respect to the Z-axis when a data conversion was completed. This bit will be set only if a 3D tablet is being used. Read only. Cleared by initiation of a new conversion or RESET.

b. Data Tablet X Register (DTXDAT) - SCB:177665

This register contains a number which represents the pen X position relative to the origin of the tablet.

c. Data Tablet Y Register (DTYDAT) - SCB:177666

This register contains a number which represents the pen Y position relative to the origin of the tablet.

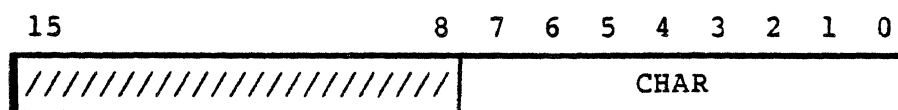
d. Data Tablet Z Register (DTZDAT) - SCB:177667

This register contains a number which represents the pen Z position relative to the origin of the 3D tablet. This register is used only when a 3D data tablet is interfaced to the system.

2.6.3 Alphanumeric Keyboard

The standard Picture System Alphanumeric Keyboard is a 61-key, 128-character keyboard. Each time a key is depressed, the associated KB REQ bit in the DEVREQ 0-15 register is set, generating an interrupt request to the Picture Controller if the associated KB IE bit is set in the DEVIE 0-15 register. (See Section 2.5 for specific interrupt control details.)

a. Keyboard Data Register (KBDATA) - SCB:177607



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
7-0	CHAR	The ASCII code for the character which was most recently typed may be read from this register. If interrupts are enabled and a keystroke occurs, a device interrupt will be generated.

2.6.4 Function Switches & Lights/Lighted Function Buttons

There are two switch options available for the Picture System. The Function Switches & Lights is a unit with 16 paddle switches, each with 3 positions (OFF/ON momentary ON) and a separate incandescent light associated with each switch. The Lighted Function Buttons is a unit with 32 back lighted buttons (1 row of 4, 4 rows of 6, and 1 row of 4). For both options, the transition of any switch or button (0 to 1 or vice versa) will cause the associated FS REQ bit to be set in the DEVREQ 0-15 register, generating an interrupt request to the Picture Controller if the associated FS IE bit is set in the DEVIE 0-15 register. (See Section 2.5 for specific interrupt control details.)

For the Function Switches & LIghts, there are two data registers, FSWR and FSLR, which provide control and status information.

a. Function Switch Register (FSWR)-SCB:177626

Bits 15-0 correspond to the setting of switches 15-0 on the function switch box. The corresponding bits are 1 if the switch is set, 0 if not set. Read only.

b. Function Switch Light Register (FSLR)-SCB:177627

Bits 15-0 correspond to the status of lights 15-0 on the function switch box. The corresponding bits are 1 if the light is on, and 0 if off. Read/Write.

For the Lighted Function Buttons (also called Extended Switches and Lights), there are eight data registers, ESWR0-ESWR3 and ESLR0-ESLR3, which provide control and status information.

c. Extended Switch Registers (ESWR0-ESWR3)

The status of the buttons may be read from the 8 least significant bits of these registers. The corresponding bits are 1 if the button has been pushed, 0 if not. Read only.

<u>Register</u>	<u>SCB</u>	<u>Buttons</u>
ESWR0	177440	7-0
ESWR1	177441	15-8
ESWR2	177442	23-16
ESWR3	177443	31-24

d. Extended Light Registers (ESLR0-ESLR3)

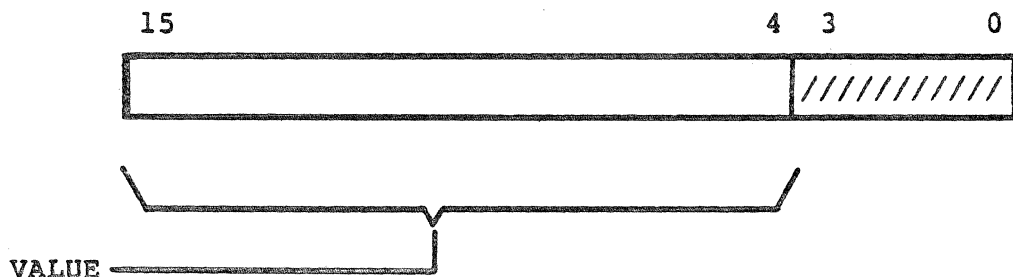
The lights may be written or their current status read using the 8 least significant bits of these registers. The corresponding bits are 1 if the light is on, 0 if off. Read/Write.

<u>Register</u>	<u>SCB</u>	<u>Lights</u>
ESLR0	177444	7-0
ESLR1	177445	15-8
ESLR2	177446	23-16
ESLR3	177447	31-24

2.6.5 Control Dials/Joystick

The Picture System Control Dials consist of a table top enclosure with eight continuous turn potentiometers. The Joystick option provides a box with a three-axis, return to center, joystick. Both of these analog devices are interfaced to the PSBUS by means of a 12-bit analog-to-digital (A/D) converter. Each A/D interface provides for the A/D conversion of 16 channels on a continuous basis at a 33 MHz rate (every channel updated approximately each 2.1 usec). A single channel is dedicated for each Control Dial and each axis of the Joystick. Thus, two Control Dial units or five Joysticks (or one Control Dial unit and two Joysticks) may be interfaced to the Picture System by a single A/D interface. There are 16 data registers, ADDR0-ADDR17, which provide the current digital value of each of the analog channels.

a. A/D Data Register (ADDR0 - ADDR17) - SCB:177500-177517



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15-4	VALUE	The current value of the A/D channel associated with this register is continually updated by the A/D interface. The value appears as a 12-bit left justified number, accurate to $\pm 1/2$ bit.

2.6.6 Light Pen

The light pen is a device for user interaction which allows identification of an object displayed on the screen. It consists of a fast photodetector which is interfaced to the Picture Generator and the PSBUS. During the refresh of a picture, if the photodetector is pointed at a line segment, character, or dot, information sufficient to identify the drawn entity is saved and made available to the application program.

The light pen is a fast (<300ns response) photodetector in a package which includes a tip switch which may be pressed against the CRT face to provide additional information to an application program. The unit is cylindrical with 0.5 inch diameter, 6 inch length and a 0.100 inch acceptance area.

The light pen interface to the Picture Generator provides several types of information to the programmer which can be related to both the picture structure and the geometry of the application program. The displayed picture can be divided into named segments which could, for instance, correspond to objects of interest to an application program. Identification of the object the light pen is pointing at would consist of reading the segment name which was saved when the pen hit occurred.

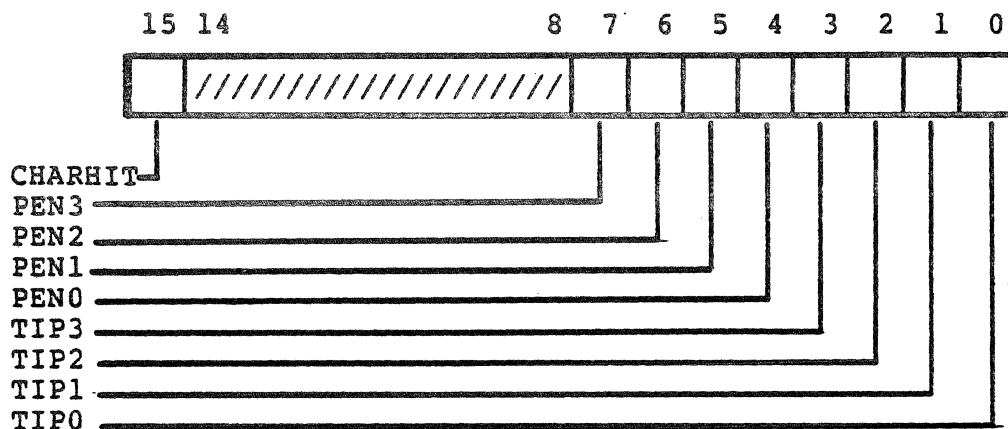
Many applications are concerned with identifying light pen hits to finer levels of detail. It is common to want to know which line segment was detected by the light pen and/or what the coordinates of the hit point were in the untransformed user data base. In an environment in which clipping occurs, in which the components of the graphics hardware are pipelined and in which the refresh memory contents are not necessarily static because of segment relocation, a refresh buffer address and correlation table is not an appropriate method of light pen interaction. By taking advantage of the digital character of the Picture Generator, one can provide x and y screen coordinate information which can be used to deduce the screen coordinates of a point which caused a pen hit. This two-dimensional digital information is handled in the same manner as similar information from a data tablet. Using hit testing in the Picture Processor, which element of a picture that caused a pen hit can be determined.

PS2/MPS Reference Manual
Hardware Details

Control of the light pen is provided by eight registers in the SCB and three bits in the Picture Generator Refresh Control Command.

The light pen is enabled and disabled by the light pen control bits in word 1 of a Picture System 2 Refresh Control Command (see Section 2.4.3). The LP TAKE bit (bit 8) is set to enable this Refresh Control Command to be taken to modify the Light Pen status. If LP TAKE is not set, then the current Light Pen status will be unmodified by the LP ENABLE and LP NAME bits. The LP ENABLE bit (bit 9) is set to enable light pen sensitivity and cleared to disable it. This allows a program to make only certain portions of a picture visible to the light pen. This prevents having to discard unwanted light pen hits and allows the Picture Generator to process short line segments at maximum speed. The LP NAME bit (bit 10) is set to identify this Refresh Control Command segment name as a light pen name. The segment name contained in word 2 of the Refresh Control Command is loaded into the Light Pen Segment Name register (see below) when this command is encountered. Once the light pen is enabled and a picture is being refreshed, information related to light pen hits can be read from the light pen registers in the SCB detailed below.

a. Light Pen Status Register (LPSR)-SCB:177720



<u>BIT</u>	<u>NAME</u>	<u>FUNCTION</u>
15	CHARHIT	Set to indicate that the light pen hit was caused by a character. Read only.

14-8	not used	
7-4	PEN3-0	Set when a hit is detected by the associated pen. Cleared by acknowledging the LPEN REQ bit in the System Interrupt Request register (SYSREQ, see Section 2.5). Read only.
3-0	TIP3-0	Set when associated light pen tipswitch is actuated and cleared when tipswitch is released. Read only.

- b. Light Pen Segment Name Register (LPSN)-SCB:177721
195271
If PEN3-0 is set in the LPSR, this register contains the segment name encountered in the last Refresh Control Command with the LP NAME bit set before the pen hit occurred. Read only.

- c. Subcount/Character Count Register (LPSCT)-SCB:177722
195273
If the CHARHIT bit is set in the LPSR, this register contains the number of characters processed by the Character Generator since the last Refresh Control Command (segment name) with the LP NAME bit set was encountered by the Picture Generator.

Subcount

If the CHARHIT bit is not set and PEN3-0 is set, this register contains a count which is proportional to the distance from the start point of the line segment which caused the pen hit to the point where the line segment entered the field of view of the light pen. Read only.

- d. Total Count Register (LPTCT)-SCB:177723

If PEN3-0 is set, this register contains a count which is proportional to the distance from the start point of the line segment which caused the pen hit to the end point of the same line segment.

The ratio of the Subcount to the Total count is a fraction which indicates where along the line the light pen detected the line segment. Use of a ratio of the Subcount and Total Count values makes it unnecessary to explicitly compensate for different Picture Generator drawing rates. Read only.

The following four registers each contain a 16-bit two's complement number which may range from -2048 to 2047 (screen coordinate values).

e. Start Point X Coordinate Register (LPSX)-SCB:177724

If PEN3-0 is set, this register contains the x screen coordinate of the starting point of the line segment which caused the pen hit. Read only.

f. Start Point Y Coordinate Register (LPSY)-SCB:177725

If PEN3-0 is set, this register contains the y screen coordinate of the starting point of the line segment which caused the pen hit. Read only.

X Character Coordinate

g. End Point X Coordinate Register (LPEX)-SCB:177726

If PEN3-0 is set, this register contains the x screen coordinate of the end point of the line segment which caused the pen hit. Read only.

If CHARHIT and PEN3-0 are set, this register contains the x screen coordinate of the character which caused the hit.

Y Character Coordinate

h. End Point Y Coordinate Register (LPEY)-SCB:177727

If PEN3-0 is set, this register contains the y screen coordinate of the end point of the line segment which caused the pen hit. Read only.

If CHARHIT and PEN3-0 are set, this register contains the y screen coordinate of the character which caused the hit.

Interrupt control is provided by the Light Pen Request (LPEN REQ) and Interrupt Enable (LPEN IE) bits in System Interrupt Control registers (see Section 2.5 for specific details). LPEN REQ is set when any pen hit or tip switch transition occurs.

The light pen is used to interact with a computer program in several ways. It can be used to point at something (an illuminated entity on the screen), to point at nothing (a blank place on the screen) or to provide a dynamic reference point on the screen (light pen tracking). "Pointing at something" is easily accomplished by inspecting the Light Pen Control Registers in the SCB at the end of a frame refresh. The first pen hit will have caused the relevant data to have been latched. The Segment Name associated with the hit item is available for high level correlation. The x and y screen coordinates of the hit point are available by using start and end point coordinates and the Subcount/Count ratio. By performing an inverse viewport mapping, which can be determined by knowing what viewport was in effect when the segment was created, one can transform the 12 bit x and y screen coordinates into a 16 bit number which can be used to detect a line segment in the users untransformed 16 bit data space by hit testing techniques currently used in conjunction with the data tablet. The amount of hit testing to be done can be limited because the segment of the picture which caused the hit is known.

A possible refinement of hit testing techniques for use with both light pen and data tablet which allows one to determine the 16 bit untransformed x,y and z coordinates of the point at which a line segment entered the hit window is accomplished using the Picture Processor. This can be done by loading the hit window, setting the Picture Processor to hold and interrupt on a hit and outputting the data to be hit tested. When a hit occurs, analysis of the MAP parameter register file contents, the hit bits and the IB bit in the MSR allow the untransformed point which corresponds to the point at which the hit occurred to be determined. This is possible because the unclipped, untransformed endpoints, the unclipped, transformed endpoints and clipped, transformed endpoints are all available. The clipped transformed endpoints can be projected onto the unclipped, untransformed line to produce a corresponding untransformed point.

"Pointing at nothing" and light pen tracking are similar in that the programmer must determine where the pen is pointing by displaying "tentative" lines on the screen and seeing if a light pen hit occurs. Both of these functions can be done by using the direct I/O path to the Picture Generator Passive Input Port at the end of frame refresh. Pen position can be determined by drawing a figure which will eventually cover the screen and testing for light pen hits. Pen tracking can be accomplished by drawing the tracking figure a stroke at a time and testing the PEN bits. Because the geometry of the tracking figure is known, the Subcount alone is sufficient to determine pen position relative to the center

of the figure. After all strokes are drawn, a new reference point may be determined.

Restrictions and limitations:

Because the light pen interface relies on the digital Picture Generator for line endpoint information, a modified timing equation must be used to determine drawing speed when the light pen is enabled. In normal operation, when the Picture Generator draws a series of short vectors ($<0.218''$), a digital endpoint value is passed to the D/A converters at the end of every other vector. By making this information available at every vector endpoint, additional time is used. When drawing short vectors with the light pen enabled, the time required to draw a line segment is determined by the following equation:

for $L < 0.218''$ draw time = $[1.6 + (2.586 * L)]$ microseconds

where L is the line length in inches.

An extra command (MOVE or STATUS) must be included at the end of a frame if the last Picture Generator command is a DRAW. This is necessary to cause the final vector endpoint to be sent to the light pen interface.

To assure proper correlation of segment names to vector information, when a Refresh Control Command is encountered with the LP NAME bit set, the Picture Generator is flushed. The time necessary to refill the Picture Generator pipeline after a light pen segment name has caused it to flush is 1.7 microseconds.

2.6.7 Remote Terminal Interface

The Remote Terminal Interface (RTI) is an option for the Picture System which enables it to function as an 80 character by 48 line* alphanumeric terminal as well as a general purpose graphics display system. The RTI interfaces the Picture System to the host computer by means of a serial communication line and an RS-232C (or 20 mA current loop) connection. The RTI utilizes the Picture Generator to actually display the characters and line positioning commands on the Picture Display. Figure 2.6-1 shows the basic interface paths of the RTI to the host computer and Picture System.

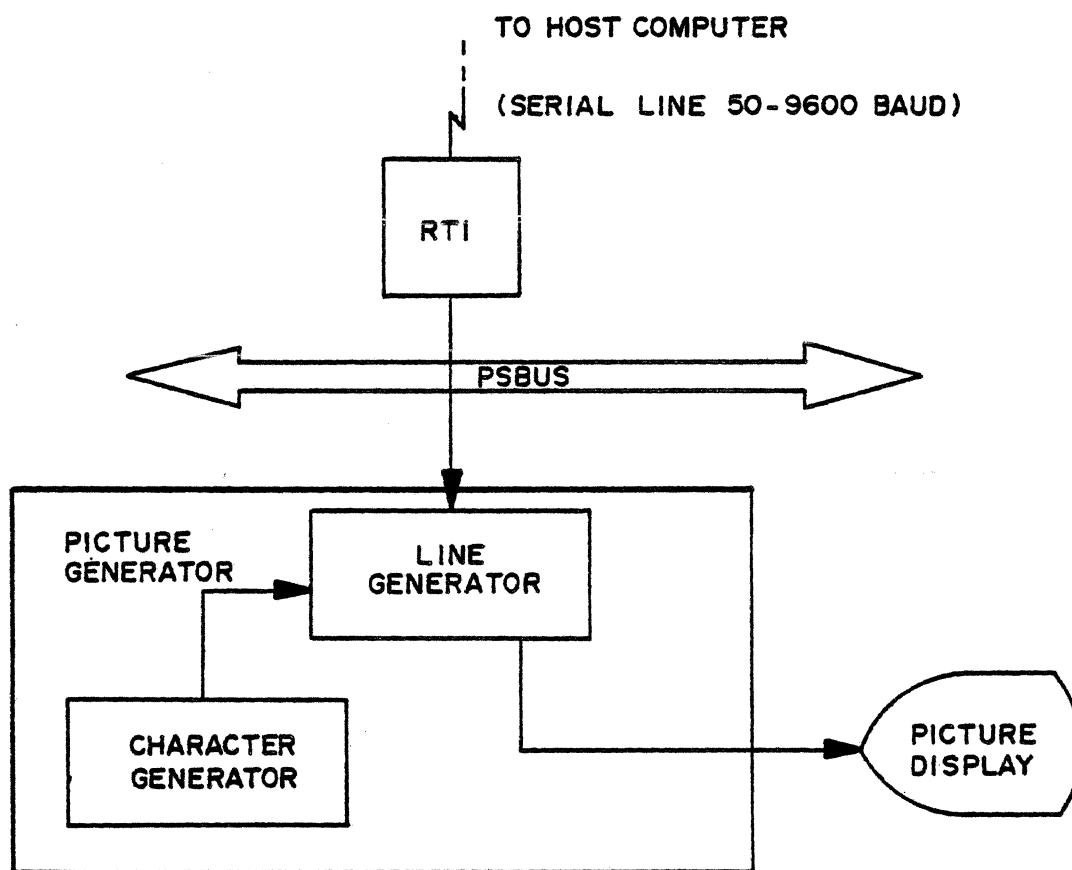


Figure 2.6-1
RTI Interface Paths

*The number of lines of text actually displayed is limited by the internal storage (2048 characters). The number of lines displayed is dynamically adjusted to a maximum of 48 lines by the RTI to maximize the use of the internal storage.

The standard RTI consists of five major components; the RTI Keyboard, RTI Universal Receiver/Transmitter (UART), RTI Control, RTI Refresh Memory and RTI/PS Interface. The RTI can also be configured to include an option* which allows it to function as an Active/Passive device on the PSBUS. These components and their logical interconnections are shown in Figure 2.6-2.

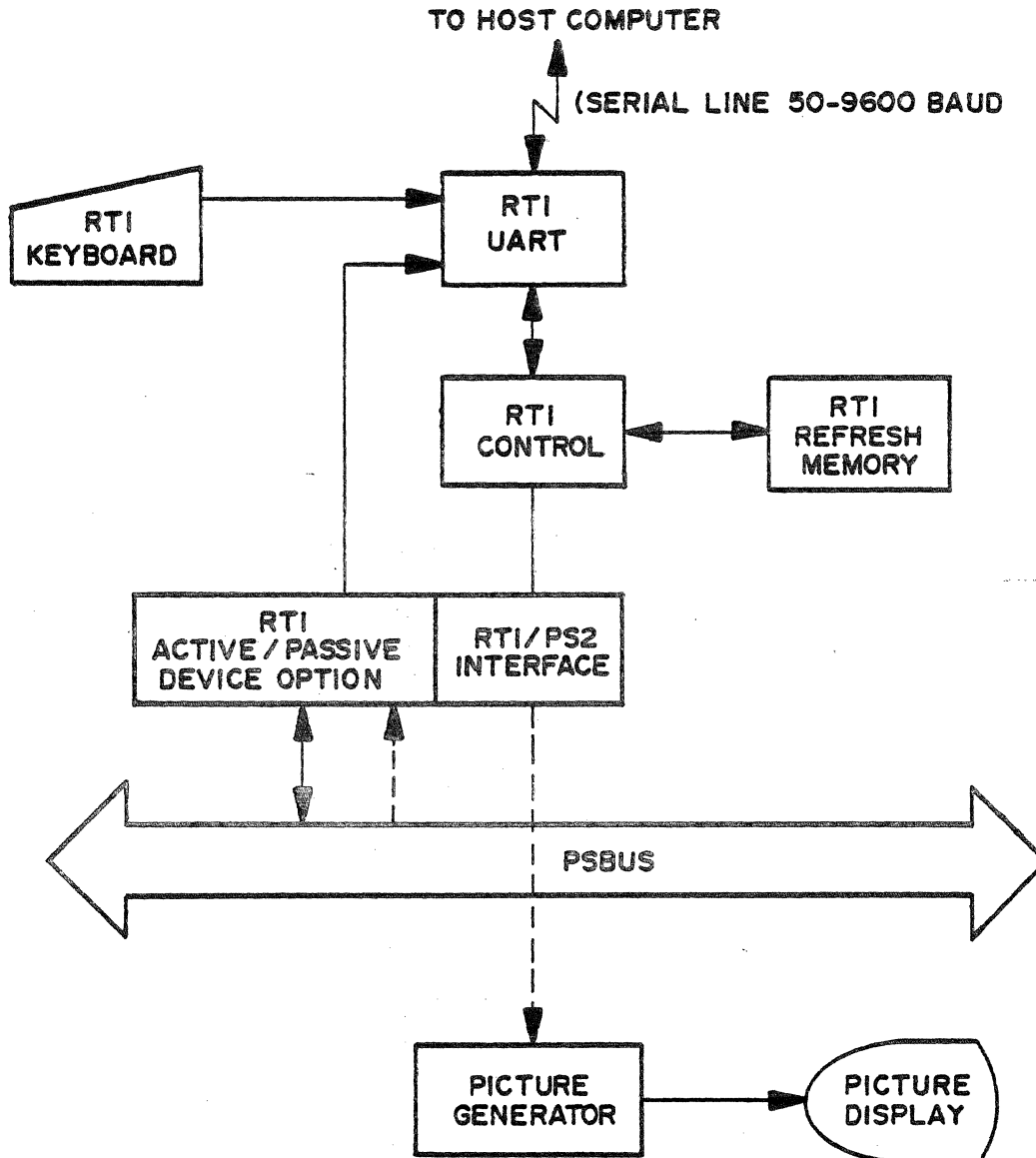


Figure 2.6-2
RTI Basic Block Diagram

*The Active/Passive RTI option is not available for customer installations.

As characters are received from the host computer by the RTI UART, they are accepted by the RTI Control and deposited in the RTI Refresh Memory for subsequent display (each character received is examined by the RTI Control and if it is one of the special function characters it is not deposited in the RTI Refresh Memory, see below). The RTI is synchronized to the 60 Hz line frequency and slaved to the Picture System 2 Refresh Controller or Multi Picture System Refresh and Device Processor. When the Refresh Controller or Refresh and Device Processor finishes a normal refresh cycle the RTI Control sends the current contents of the RTI Refresh Memory to the Picture Generator via the PG Passive Input Port (PGPIP). When the output of the RTI Refresh Memory is completed, the Refresh Controller is then allowed to issue the RFSTOP interrupt request (RFSTOP REQ; see Section 2.5.1.2) or the Refresh and Device Processor is allowed to issue the RP interrupt request (RP REQ; see Section 2.5.2). In this manner the RTI is able to "share" the Picture Generator without software control or intervention. Thus, the RTI is totally transparent to the normal operation of the system. The RTI was specifically designed to be an independent entity within the Picture System and which could not be affected by the manner in which the Picture System was programmed. If there is no currently executing graphics program (i.e., the Refresh Controller/Refresh and Device Processor is not operating) then the RTI is merely synchronized to the 60 Hz line frequency and operates in the same manner.

The Keyboard used with the Remote Terminal Interface is the standard keyboard available for the Picture System (see Section 1.6.4). There are however, three keys* on the keyboard which have special functions when used in conjunction with the RTI. Striking the "CLEAR" key causes the RTI to reset to its initial state clearing the contents of the Refresh Memory such that no characters are displayed, no cursor (a current line/character indicator) is displayed and the RTI does not interact with the Picture Generator at all. Thus, this key enables the Picture System to be used exclusively for graphics display. This condition remains in effect until a character is received from the host computer. The "REM/LOCAL" key is an alternate action key which when in the LOCAL position disables the RTI interface and enables the keyboard to be used in conjunction with the standard Picture System keyboard interface (see Section 2.6.3). Thus, the keyboard may be used in both ways under direct user control. If the keyboard is not connected to a Picture System Keyboard Interface (E&S Part # 195171-100) then the "REM/LOCAL"

*Not available on earlier models of the Picture System standard keyboard.

key provides a half/full duplex capability for the RTI. Striking the "BREAK" key causes the RTI to issue a break signal to the host computer (the transmission line is held in the SPACE condition).

The RTI Universal Receiver/Transmitter (UART) and associated electronics perform the packing/unpacking of the serial data for the transmission to/from the host computer. It has the ability to be installed such that characters may be sent or received at the following baud rates: 50, 75, 100, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200 or 9600. The baud rate that the RTI is to function at is selectable by dip switch settings on the RTI Control card (E&S #195175-100), IC location 24. The setting of the dip switches for each of the baud rates available is shown in Table 2.6-1.

Table 2.6-1
RTI Baud Rate Switch Settings

SWITCH				BAUD	SWITCH				BAUD
1	2	3	4	Send	5	6	7	8	Recv
x	x	x	x	50	x	x	x	x	50
x	x	x	0	75	x	x	x	0	75
x	x	0	x	110	x	x	0	x	110
x	x	0	0	134.5	x	x	0	0	134.5
x	0	x	x	150	x	0	x	x	150
x	0	x	0	300	x	0	x	0	300
x	0	0	x	600	x	0	0	x	600
x	0	0	0	1200	x	0	0	0	1200
0	x	x	x	1800	0	x	x	x	1800
0	x	x	0	2000	0	x	x	0	2000
0	x	0	x	2400	0	x	0	x	2400
0	x	0	0	3600	0	x	0	0	3600
0	0	x	x	4800	0	0	x	x	4800
0	0	x	0	7200	0	0	x	0	7200
0	0	0	x	9600	0	0	0	x	9600

(OPEN=0)

The RTI UART may be connected to the host computer serial interface by either a RS-232C or 20 mA current loop connection.

The RTI Control is the component of the RTI that manages the storing of characters received into the RTI Refresh Memory and the display of the characters via the Picture Generator. The RTI Control examines each character as it is received and if it is not one of the set of special codes (see Table 2.6-2) then it stores it directly in the RTI Refresh Memory. If it is a special code, then it performs the appropriate function. The RTI Control manages the RTI Refresh Memory such that the RTI/PS appears to be a 48 line, 80 character/line scrolling alphanumeric terminal. When the RTI is used in conjunction with a refreshing graphics program, the RTI automatically displays only the last four lines in the RTI Refresh Memory (compressed mode) leaving the major portion of the screen free to display graphical data. Prior to each RTI refresh cycle, the RTI Control outputs data to the Picture Generator which serves to initialize the Line Generator and Character Generator to a known state. The data output are:

```

000300 Status Command (with CGRESET)
176000 Scope Select (dip switch selectable)
040600 PS2 Refresh Control Command (light pen disable)
000000 Null Segment Name
134000 Move Command (X=-2048, Z=77)
173720 (Y=2000) or 174420 (Y=-1776) for compressed mode
000003 Characters (Push-Load Font Stack)
001004 (character size=4, HOME)
000000
CH1CH0
:
: Character Data (RTI Refresh Memory)*
:
000000
CHnCH
002036 Characters (Cursor, Pop Font Stack)
000000 (null characters)
100000 Move Command (X=0, Z=0)
000000 (Y=0)
100000 Move Command (X=0, Z=0)
000000 (Y=0)

```

The current Line Generator/Character Generator status is destroyed by this operation thereby constraining each user buffer of refresh commands in Picture Memory to have initial Line Generator/Character Generator status commands to ensure the integrity of the status of the Picture Generator.

*Note that the characters output from the RTI Refresh Memory are always output as Word 2 of the Line Generator Character Command. This enables full 8-bit character codes to be transmitted and displayed by the RTI.

The RTI Control is the component of the RTI that manages the storing of characters received into the RTI Refresh Memory and the display of the characters via the Picture Generator. The RTI Control examines each character as it is received and if it is not one of the set of special codes (see Table 2.6-2) then it stores it directly in the RTI Refresh Memory. If it is a special code, then it performs the appropriate function. The RTI Control manages the RTI Refresh Memory such that the RTI/PS appears to be a 48 line, 80 character/line scrolling alphanumeric terminal. The RTI has a dip switch selectable option which places it in COMPRESSED mode. COMPRESSED mode used in conjunction with a graphics program displays only the last four lines in the RTI refresh memory. RTI's using 195175-102/103 cards also have a jumper selectable option which allows the "Break Key" to toggle the RTI display between "Full Screen" and "Compressed Mode". Once in compressed mode all lines in the RTI refresh memory except the last four are lost. Toggling back to full screen mode will produce only the last four lines.

Prior to each RTI refresh cycle, the RTI Control outputs data to the Picture Generator which serves to initialize the Line Generator and Character Generator to a known state. The data output are:

```

000300 Status Command (with CGRESET)
176000 Scope Select (dip switch selectable)
040600 PS2 Refresh Control Command (light pen disable)
000000 Null Segment Name
134000 Move Command (X=-2048, Z=77)
173720 (Y=2000) or 174420 (Y=-1776) for compressed mode
000003 Characters (Push-Load Font Stack)
001004 (character size=4, HOME)
000000
CH1CH0
.
.
.
000000
CHnCH
002036 Characters (Cursor, Pop Font Stack)
000000 (null characters)
100000 Move Command (X=0, Z=0)
000000 (Y=0)

```

} Character Data (RTI Refresh Memory)*

*Note that the characters output from the RTI Refresh Memory are always output as Word 2 of the Line Generator Character Command. This enables full 8-bit character codes to be transmitted and displayed by the RTI.

The RTI Refresh Memory is a 2048 x 8-bit memory (distinct from the Picture Memory) that is used to buffer the characters to be displayed. This memory is treated as a circular memory with the "oldest" line scrolled out of the memory when a new line is output to a currently full "page". The end of a line is determined by the presence of a Line Feed (LF, 012) character.

The RTI/PS Interface provides the path from the RTI Control/Refresh Memory to the Passive Input Port of the Picture Generator. This is not a standard PS interface to the PSBUS since only the PGPIP may be written by the RTI/PS Interface.

The RTI Active/Passive Device option is an additional Picture System card that allows the RTI to function as a general purpose Active/Passive device on the PSBUS. This means that any location in PSMEM or the SCB may be read or written using this path. When this option is installed, three additional "special codes" are enabled for the RTI control. These special codes (DC2, DC3, DC4; see Table 2.6-2) allow blocks of data to be read or written using the RTI as well as a PSRESET to be issued. It should be noted that the use of the Active/Passive device option requires that the serial interface to the host computer be capable of sending and receiving a full 8-bit character code without the high-order bit having parity or other meaning. Similarly, the operating system through which communication with the RTI is accomplished must be able to read/write character data without interpretation (i.e., binary format).

Since the RTI utilizes the Character Generator to actually display the text on the Picture Display, all of the characters described in Section 2.4.4 may be output to the RTI for display. However, because the RTI Control examines each character as it is received from the host computer, the user should restrict output to the RTI to be the 95 displayable ASCII characters defined in Table 2.4-4 and the special codes interpreted by the RTI and defined in Table 2.6-2.

The current Line Generator/Character Generator status is destroyed by this operation thereby constraining each user buffer of refresh commands in Picture Memory to have initial Line Generator/Character Generator status commands to ensure the integrity of the status of the Picture Generator.

The RTI Refresh Memory is a 2048 x 8-bit memory (distinct from the Picture Memory) that is used to buffer the characters to be displayed. This memory is treated as a circular memory with the "oldest" line scrolled out of the memory when a new line is output to a currently full "page". The end of a line is determined by the presence of a Line Feed (LF, 012) character.

The RTI/PS Interface provides the path from the RTI Control/Refresh Memory to the Passive Input Port of the Picture Generator. This is not a standard PS interface to the PSBUS since only the PGPIP may be written by the RTI/PS Interface.

The RTI Active/Passive Device option is an additional Picture System card that allows the RTI to function as a general purpose Active/Passive device on the PSBUS. This means that any location in PSMEM or the SCB may be read or written using this path. When this option is installed, three additional "special codes" are enabled for the RTI control. These special codes (DC2, DC3, DC4; see Table 2.6-2) allow blocks of data to be read or written using the RTI as well as a PSRESET to be issued. It should be noted that the use of the Active/Passive device option requires that the serial interface to the host computer be capable of sending and receiving a full 8-bit character code without the high-order bit having parity or other meaning. Similarly, the operating system through which communication with the RTI is accomplished must be able to read/write character data without interpretation (i.e., binary format).

Since the RTI utilizes the Character Generator to actually display the text on the Picture Display, all of the characters described in Section 2.4.4 may be output to the RTI for display. However, because the RTI Control examines each character as it is received from the host computer, the user should restrict output to the RTI to be the 95 displayable ASCII characters defined in Table 2.4-4 and the special codes interpreted by the RTI and defined in Table 2.6-2.

Table 2.6-2
RTI Special Codes and Functions

<u>ASCII CODE</u>	<u>MNEMONIC</u>	<u>FUNCTION</u>
007	BEL	Bell. Produces an audible tone to be issued from the keyboard. This character is not stored in the RTI Refresh Memory.
010	BS	Backspace. Causes the current character position to be updated to the previous character position and that character is then deleted from the RTI Refresh Memory. This character is not stored in the RTI Refresh Memory.
011	HT	Horizontal Tab. Causes the current character position to be updated to the next tab stop on the line. Tab stops are defined to be at the following space (or column) positions: 8, 16, 24, 32, 40, 48,....
012	LF	Line Feed. Causes the current character position to be updated to the next line without modifying the relative position within the line. Used to determine the end of line with the RTI Refresh Memory.
014	FF	Form Feed. Causes the current character position to be updated to the current top and left margin position and the screen to be cleared. Logically equivalent to the striking of the "CLEAR" key on the keyboard. This character is not stored in the RTI Refresh Memory.
015	CR	Carriage Return. Causes the current character position to be updated to the left margin

021 DC1

position.

Device Code 1 (Line Generator Commands). The next character received after the DC1 is interpreted to be the two's complement count of the number of characters/2 to be deposited directly into the RTI Refresh Memory without interpretation by the RTI Control. This provides the capability of outputting Line Generator Commands (e.g., MOVE or DRAW, etc.; see Section 2.4.3) via the RTI. For example, to output the Line Generator commands to draw a box around the edges of the screen, the following 22 character codes would be output:
021, 365, 267, 377, 365, 377,
367, 377, 370, 0, 370, 0, 370,
0, 370, 377, 365, 377, 367,
377, 365, 377.

These would be interpreted to be the following:

DC1, -22
133777,173777
(MOVE; X=3777, Y=3777, Z=77)
173777,174000
(DRAW; X=3777, Y=4000, Z=77)
174000,174000
(DRAW; X=4000, Y=4000, Z=77)
174000,173777
(DRAW; X=4000, Y=3777, Z=77)
173777,173777
(DRAW; X=3777, Y=3777, Z=77)

022 DC2

Device Code 2 (RTI Active Output). This code, valid only when the RTI Active/Passive device option is available, functions such that the next character received after the DC2 is interpreted to be the two's complement count of the number of characters/2 to be transferred to the Picture System. The next two characters are assembled to form the 16-bit address within the

PSMEM or SCB to write the data. This provides the capability to transfer data directly to PSMEM (or SCB registers). The data in PSMEM could then be refreshed using the autorefresh mode of the Refresh Controller. For example, to output the Line Generator Commands of the example shown for DC1 to PSMEM starting at location 0, the following 24 character codes would be output:

022, 363, 0, 0, 267, 377, 365,
377, 367, 377, 370, 0, 370, 0,
370, 0, 370, 377, 365, 377,
367, 377, 365, 377.

These would be interpreted to be the following:

DC2, -24
0 (PSMEM location 0)
133777,173777
(MOVE; X=3777, Y=3777, Z=77)
173777,174000
(DRAW; X=3777, Y=4000, Z=77)
174000,174000
(DRAW; X=4000, Y=4000, Z=77)
174000,173777
(DRAW; X=4000, Y=3777, Z=77)
173777,173777
(DRAW; X=3777, Y=3777, Z=77)

023

DC3

Device Code 3 (RTI Active Input). This code, valid only when the RTI Active/Passive device option is available, functions such that the next character received after the DC3 is interpreted to be the two's complement count of the number of characters/2 to be transferred from the Picture System. The next two characters are assembled to form the 16-bit address within PSMEM or the SCB from which data are to be read. The data read beginning at this address are sent to the host computer as 8-bit characters, each two characters forming a 16-bit

word of data. For instance, the following characters would use this command to read the switch register of the Function Switches & Lights option: 023, 377, 377, 210.

When the beginning address to be read is received by the RTI Control, the data will be read via the PSBUS and transferred back to the host computer as 8-bit characters. If the value in the switch register is 123456, the characters returned for the above command sequence are:
56, 247.

024 DC4

Device Code 4 (PSRESET). This code, valid only when the RTI Active/Passive device option is available, causes a PSRESET signal (see Section 2.2) to be issued. This PSRESET signal lasts for 100 msec and characters received by the RTI during this time are ignored.

177 DEL

Delete. Causes the current character position to be updated to the previous character position and that character is then deleted from the RTI Refresh Memory. This character is not stored in the RTI Refresh Memory.

NOTE

Whenever the PSRESET signal is issued (either by the DC4 code or the Picture Controller Interface) the RTI is in a reset state for 100 msec. All characters received by the RTI during this time are ignored.

The RTI Active/Passive Device option provides the capability of the RTI to function as a Passive device. When utilized in this manner, all data directed to its Passive Input Port (RTIPIP, SCB:177772) are converted to serial data and transmitted to the host computer as 8-bit character codes.

a. RTI Passive Input Port (RTIPIP)-SCB:177772

The RTI Passive Input Port is a 16-bit write-only location in the Picture System SCB to which data may be directed by an Active device for transmission to the host computer. In order to use the RTIPIP, the Active device's output address register should be set to address the RTIPIP using the Direct I/O path to the Picture System. Once this has been done, all subsequent data output by the Active device will be transmitted to the host as 8-bit character codes, at the baud rate that the RTI "Recv" is set to (see Table 2.6-1).

2.6.8 User Devices

All devices previously detailed throughout this chapter are for the standard Picture System. There are, however, certain instances in which special devices are required for a given application. The architecture of Picture System facilitates the interfacing of these "user devices" by providing 18 device slots in each Picture System backpanel into which standard and user devices may be inserted. Each device slot is pre-wired to provide all interfacing signals (i.e., interrupt control, address selection, etc.) necessary to interface a device.

There are three types of devices:

1. Passive Devices

(Real-time Clock, Data Tablet, Alphanumeric Keyboard, Function Switches & Lights, Light Pen, Control Dials/Joystick)

2. Passive Devices with a Passive Input or Output Port

(Picture Controller Interface, Line Generator Input Controller, MAP Output Formatter, MAP Input Controller)

3. Active Devices

(Picture Controller Interface, Refresh Controller or Refresh and Device Processor, MAP Output Formatter, MAP Input Controller, Remote Terminal Interface)

These types of devices are listed in increasing order of complexity. Type 1 typically requires only 1 device slot; types 2 and 3 usually require 2 device slots. Types 2 and 3 are more complex as they require interfacing to one of the eight Picture System data channels of the PSBUS. Each of these data channels are capable of supporting a Passive Device with a Passive Input or Output Port and an Active Device. Thus, there may be a maximum of 8 Type 2 devices and 8 Type 3 devices. Other than device slot allocation in the Picture System backpanel, there is no maximum number of Type 1 devices.

PS2/MPS Reference Manual
Hardware Details

The Picture System has the following standard device assignments:

<u>Device Name</u>	<u>Device Type</u>	<u>Number of Device Slots</u>	<u>PSBUS Data Channel Assignment</u>
PS2 Real-Time Clock	1	1	none
Data Tablet	1	1	none
Alphanumeric Keyboard	1	1	none
Function Switches & Lights	1	1	none
Lighted Function Buttons	1	1	none
Control Dials/Joystick	1	2	none
Light Pen	1	1	none
Picture Controller Interface	2,3	2	DIO-4 (Type 3) DMA-0 (Type 2,3)
LG Input Controller	2	0*	5
Refresh Controller	3	0*	5
Refresh and Device Processor	3	0*	5
MAP Output Formatter	2,3	0*	6
MAP Input Controller	2,3	0*	7
RTI (standard)	3	2	2
RTI (optional)	2,3	3	2

The standard Picture System utilizes 2 of the 18 available device slots and 5 of the 8 available PSBUS data channels. This provides 16 device slots and 3 PSBUS data channels available for configuring Picture System devices on a Picture System 2 or Multi Picture System.

*These devices are interfaced to special device slots in the Picture System backpanel and therefore do not require one of the 18 standard device slots.

2.7 Map of the System Control Block (SCB)

The System Control Block contains all the Picture System device control and maintenance registers as detailed throughout this chapter. Table 2.7-1 is a map of all locations in the SCB that are assigned or reserved for Picture System use.

PS2/MPS Reference Manual
Hardware Details

Table 2.7-1
System Control Block

<u>Page</u>	<u>From</u>	<u>To</u>	<u>Device</u>	<u>Register Name</u>	<u>Remarks</u>
	177400-177437		unused		
2-239	177440		ESL0	ESWR0	Lighted Function Buttons (Extended Switches and Lights)
2-239	177441		ESL0	ESWR1	
2-239	177442		ESL0	ESWR2	
2-239	177443		ESL0	ESWR3	
2-239	177444		ESL0	ESLR0	
2-239	177445		ESL0	ESLR1	
2-239	177446		ESL0	ESLR2	
2-239	177447		ESL0	ESLR3	
	177450-177457		ESL1		
	177460-177467		ESL2		
	177370-177477		ESL4		
2-240	177500		A/D0	ADDR0	Control Dials/Joystick
2-240	177501		A/D0	ADDR1	
2-240	177502		A/D0	ADDR2	
2-240	177503		A/D0	ADDR3	
2-240	177504		A/D0	ADDR4	
2-240	177505		A/D0	ADDR5	
2-240	177506		A/D0	ADDR6	
2-240	177507		A/D0	ADDR7	
2-240	177510		A/D0	ADDR10	
2-240	177511		A/D0	ADDR11	
2-240	177512		A/D0	ADDR12	
2-240	177513		A/D0	ADDR13	
2-240	177514		A/D0	ADDR14	
2-240	177515		A/D0	ADDR15	
2-240	177516		A/D0	ADDR16	
2-240	177517		A/D0	ADDR17	
	177520-177537		A/D1		
	177540-177557		A/D2		
	177560-177577		A/D3		
	177600		KB7	KBDATA	Keyboards
	177601		KB6		
	177602		KB5		
	177603		KB4		
	177604		KB3		
	177605		KB2		
	177606		KB1		
2-237	177607		KB0		

Table 2.7-1
System Control Block

Page	From	To	Device	Register Name	Remarks
2-8	177400		MEMORY	MEMDMA	DMA Mapping Register
2-8	177404		MEMORY	MEMDIO	DIO Mapping Register
2-8	177405		MEMORY	MEMRFC	Refresh " "
2-8	177406		MEMORY	MEMMOT	MAP output " "
2-8	177407		MEMORY	MEMMIN	MAP input " "
2-8	177410		MEMORY	EXMSR	Extended Memory Status Memory
2-239	177411-177437	unused			
2-239	177440		ESL0	ESWR0	Lighted Function Buttons (Extended Switches and Lights)
2-239	177441		ESL0	ESWR1	
2-239	177442		ESL0	ESWR2	
2-239	177443		ESL0	ESWR3	
2-239	177444		ESL0	ESLR0	
2-239	177445		ESL0	ESLR1	
2-239	177446		ESL0	ESLR2	
2-239	177447		ESL0	ESLR3	
	177450-177457		ESL1		
	177460-177467		ESL2		
	177370-177477		ESL4		
2-240	177500		A/D0	ADDR0	Control Dials/Joystick
2-240	177501		A/D0	ADDR1	
2-240	177502		A/D0	ADDR2	
2-240	177503		A/D0	ADDR3	
2-240	177504		A/D0	ADDR4	
2-240	177505		A/D0	ADDR5	
2-240	177506		A/D0	ADDR6	
2-240	177507		A/D0	ADDR7	
2-240	177510		A/D0	ADDR10	
2-240	177511		A/D0	ADDR11	
2-240	177512		A/D0	ADDR12	
2-240	177513		A/D0	ADDR13	
2-240	177514		A/D0	ADDR14	
2-240	177515		A/D0	ADDR15	
2-240	177516		A/D0	ADDR16	
2-240	177517		A/D0	ADDR17	
	177520-177537		A/D1		
	177540-177557		A/D2		
	177560-177577		A/D3		
	177600		KB7	KBDATA	Keyboards
	177601		KB6		
	177602		KB5		
	177603		KB4		
	177604		KB3		
	177605		KB2		
	177606		KB1		
2-237	177607		KB0		

PS2/MPS Reference Manual
Hardware Details

Page	From	To	Device	Register Name	Remarks
	177610-177611		FSL7		Function Switches & Lights
	177612-177613		FSL6		
	177614-177615		FSL5		
	177616-177617		FSL4		
	177620-177621		FSL3		
	177622-177623		FSL2		
	177624-177625		FSL1		
2-238	177626		FSL0	FSWR	
2-238	177627		FSL0	FSLR	
	177630-177633		DT7		Data Tablets
	177634-177637		DT6		
	177640-177643		DT5		
	177644-177647		DT4		
	177650-177653		DT3		
	177654-177657		DT2		
	177660-177663		DT1		
2-233	177664		DT0	DTSR	
2-236	177665		DT0	DTXDAT	
2-236	177666		DT0	DTYDAT	
2-236	177667		DT0	DTZDAT	
	177670-177677		RF1	same as RF0	Refresh Controller
	177670-177677		RF1	same as RF0	Refresh Processor
	177700-177703		PG1	same as PG0	Picture Generator
	177704-177707		unused		
	177710-177717		PP1	same as PP0	Picture Processor
2-242	177720		LP0-3	LPSR	Light Pens
2-243	177721		LP0-3	LPSN	
2-243	177722		LP0-3	LPSCT	
2-243	177723		LP0-3	LPTCT	
2-244	177724		LP0-3	LPSX	
2-244	177725		LP0-3	LPSY	
2-244	177726		LP0-3	LPEX	
2-244	177727		LP0-3	LPEY	
2-78	177730		RF0	RFCSN	Refresh Controller
2-78	177731		RF0	RFSN	
2-78	177732		RF0	RFAWA	
2-79	177733		RF0	RFAWL	
2-79	177734		RF0	RFAIA	
2-79	177735		RF0	RFASA	
2-79	177736		RF0	RFAIL	
2-80	177737		RF0	RHSR	
2-86	177730		RP0	RFDT	Refresh and Device Processor
2-89	177731		RP0	RF	
2-90	177732		RP0	RF	
2-91	177733		RP0	RF	
2-93	177734		RP0	RF	
2-94	177735		RP0	RF	
	177736-177737		unused by Refresh and Device Processor		

PS2/MPS Reference Manual
Hardware Details

<u>Page</u>	<u>From</u>	<u>To</u>	<u>Device</u>	<u>Register Name</u>	<u>Remarks</u>
	177610-177611		FSL7		Function Switches & Lights
	177612-177613		FSL6		
	177614-177615		FSL5		
	177616-177617		FSL4		
	177620-177621		FSL3		
	177622-177623		FSL2		
	177624-177625		FSL1		
2-238	177626		FSL0	FSWR	
2-238	177627		FSL0	FSLR	
	177630-177633		DT7		Data Tablets
	177634-177637		DT6		
	177640-177643		DT5		
	177644-177647		DT4		
	177650-177653		DT3		
	177654-177657		DT2		
	177660-177663		DT1		
2-233	177664		DT0	DTSR	
2-236	177665		DT0	DTXDAT	
2-236	177666		DT0	DTYDAT	
2-236	177667		DT0	DTZDAT	
	177670-177677		RF1	same as RF0	Refresh Controller
	177670-177677		RP1	same as RP0	Refresh Processor
	177700-177703		PG1	same as PG0	Picture Generator
	177704-177707		unused		
	177710-177717		PP1	same as PP0	Picture Processor
2-242	177720		LP0-3	LPSR	Light Pens
2-243	177721		LP0-3	LPSN	
2-243	177722		LP0-3	LP SCT	
2-243	177723		LP0-3	LPTCT	
2-244	177724		LP0-3	LPSX	
2-244	177725		LP0-3	LPSY	
2-244	177726		LP0-3	LPEX	
2-244	177727		LP0-3	LPEY	
2-78	177730		RF0	RFCSN	Refresh Controller
2-78	177731		RF0	RFSN	
2-78	177732		RF0	RFAWA	
2-79	177733		RF0	RFAWL	
2-79	177734		RF0	RFAIA	
2-79	177735		RF0	RFASA	
2-79	177736		RF0	RFAIL	
2-80	177737		RF0	RFSR	
2-86	177730		RP0	RFDT	Refresh and Device Processor
2-89	177731		RP0	RF	
2-90	177732		RP0	RF	
2-91	177733		RP0	RF	
2-93	177734		RP0	RF	
2-94	177735		RP0	RF	
	177736-177737		unused	by Refresh and Device Processor	

PS2/MPS Reference Manual
Hardware Details

<u>Page</u>	<u>From</u>	<u>To</u>	<u>Device</u>	<u>Register Name</u>	<u>Remarks</u>
2-206	177740		PG0	PGSR	Picture Generator
2-207	177741		PG0	PGXBUS	
2-209	177742		PG0	PGYBUS	
	177743		unused		
2-230	177744		RTC	RTCCNT	PS2 Real-Time Clock
2-231	177745		RTC	RTCSR	
	177746		DMA1	DMAPSA	DMA1 PS Address
2-18	177747		DMA0	DMAPSA	DMA0 PS Address
2-57	177750		PP0	MAOL	Picture Processor
2-57	177751		PP0	MAOA	
2-29	177752		PP0	MAIA	
2-22	177753		PP0	MSR	
2-68	177754		PP0	MMSR	
2-71	177755		PP0	MMRSR	
2-72	177756		PP0	MMPAR	
2-72	177757		PP0	MMBUS	
2-216	177760		RTC	RTCREQ	PS2 Real-Time Clock
2-216	177761		RTC	RTCIE	Interrupt
2-218	177762		System	SYSREQ	PS2 System Interrupts
2-218	177763		System	SYSIE	
2-225	177762		System	SYSREQ	MPS System Interrupts
2-225	177763		System	SYSIE	
2-222	177764		Devices	DEVREQ 0-15	Device Interrupts
2-223	177765		Devices	DEVIE 0-15	
2-224	177766		Devices	DEVREQ 16-31	
2-224	177767		Devices	DEVIE 16-31	
2-19	177770		DMA0	DMAPIP	DMA Passive Input Port
	177771		reserved for options using	Passive Ports	Passive Ports
2-257	177772		RTI	RTIPIP	RTI Passive Input Port
	177773-177774		reserved for options using	Passive Ports	Passive Ports
2-76	177775		PG0	LGPIP	LG Passive Input Port
2-57	177776		PP0	MPOP	MAP Passive Output Port
2-30	177777		PP0	MPIP	MAP Passive Input Port

CHAPTER 3

3. PICTURE SYSTEM SOFTWARE CONVENTIONS

The Picture System is a highly modular computer graphics system which may be used in a variety of configurations and applications. For all configurations, however, a Picture Controller or host computer is required to initiate and control the basic operations of the Picture System. This is true for both the Picture System 2 and Multi Picture System. Although the control software in the Picture Controller may be radically different from installation to installation, several software conventions have been adopted to simplify the development and debugging of software for the Picture System. These conventions should be followed unless just cause can be found for violating a convention. Such instances must be clearly documented.

1. Interrupt processing uses the DIO path only. Interrupt service routines are not allowed to perform DMA data transfers.
2. The DIOPSA register should be read only by interrupt service routines, unless all Picture System interrupt levels are masked.
3. Passive Output Ports (i.e., MPOP) must not be addressed using the DIO path unless all Picture System interrupt levels are masked. This is necessary since data from a Passive Output Port can be read only once; therefore, data could be lost if an interrupt occurred before all data was read.
4. The Picture Processor should not be used during interrupt processing. There is no guarantee that all of the user data will have been transferred to the MAP Input Controller when the interrupt occurs.

CHAPTER 4

4. THE 4x4 MATRIX TRANSFORMATIONS USED BY THE PICTURE SYSTEM

All transformations used by the Picture System consist of simple or compound linear transformations and are implemented using 4x4 matrices. This chapter details the matrices created by the Graphics Software Packages provided for the Picture System 2 and Multi Picture System. Each subroutine calling sequence is shown, and the manner in which the parameters passed to the subroutines are used to create the transformations is indicated. In most of the subroutines (all except where the transformation is purely rotational), the inclusion of the homogeneous coordinate (W) is optional and, if not specified, is assumed to be 32767 (the largest expressable integer in the 16-bit word length data format of the Picture Processor).

4.1 Picture System 2 Graphics Software Package

The Picture System 2 Graphics Software Package provides eleven subroutines which generate 4x4 matrix transformations. Those subroutines are:

- GETROT
- GETSCL
- GETTRN
- HITWIN
- INST
- MASTER
- PSINIT
- ROT
- SCALE
- TRAN
- WINDOW

The use of these subroutines is discussed in the Picture System 2 User's Manual. This section details the 4x4 matrix transformations generated by these subroutines.

GETROT

CALL GETROT(IARRAY,ANGLE,AXIS)

One of three Matrices is produced, dependent upon the axis of rotation:

$$\text{X axis} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & A & B & 0 \\ 0 & -B & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\text{Y axis} = \begin{bmatrix} A & 0 & -B & 0 \\ 0 & I & 0 & 0 \\ B & 0 & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\text{Z axis} = \begin{bmatrix} A & B & 0 & 0 \\ -B & A & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

I = 2**14
A = I*cos(ANGLE)
B = I*sin(ANGLE)

GETSCL

CALL GETSCL(IARRAY,X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} X & 0 & 0 & 0 \\ 0 & Y & 0 & 0 \\ 0 & 0 & Z & 0 \\ 0 & 0 & 0 & W \end{bmatrix}$$

GETTRN

CALL GETTRN(IARRAY,X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ X & Y & Z & W \end{bmatrix}$$

HITWIN

CALL HITWIN(X,Y,S[,W])

A single Matrix is produced:

$$\begin{bmatrix} W & 0 & 0 & -X \\ 0 & W & 0 & -Y \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & S \end{bmatrix}$$

NOTE: This Matrix is actually the transpose of the hit window matrix. This matrix is concatenated with the current Transformation Matrix in a manner such that both are treated as transposed matrices. This performs a matrix pre-concatenation (rather than post-concatenation as is typically the case). See Section 2.3.2.2, "MATCON" for specific details.

INST

CALL INST(IL,IR,IB,IT[,IH,IY[,W]])

A single Matrix is produced:

$$\begin{bmatrix} A & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & E & 0 \\ B & D & IH & W \end{bmatrix}$$

Where:

A = (IR-IL)/2
B = (IR+IL)/2
C = (IT-IB)/2
D = (IT+IB)/2
E = IY-IH
IH = 0 default
IY = W default

MASTER

CALL MASTER(ML,MR,MB,MT[,MH,MY[,W]])

Two Matrices are produced: $\begin{bmatrix} \text{MASTER} \end{bmatrix} = \begin{bmatrix} \text{M2} \end{bmatrix} \begin{bmatrix} \text{M1} \end{bmatrix}$

$$\begin{bmatrix} 1/A & 0 & 0 & 0 \\ 0 & 1/C & 0 & 0 \\ 0 & 0 & 1/E & 0 \\ 0 & 0 & 0 & 1/W \end{bmatrix}$$

$$\begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ -B & -D & -MH & W \end{bmatrix}$$

Where:

A = (MR-ML)/2
B = (MR+ML)/2
C = (MT-MB)/2
D = (MT+MB)/2
E = MY-MH
MH = 0 default
MY = W default

PSINIT

CALL PSINIT()

The Transformation Matrix stack is reset and the Transformation Matrix is loaded with:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

$$I = 2^{*}14$$

The value of 2**14 (40000 octal) is chosen for the unity element of the identity matrix rather than 1 to maintain the most precision within the matrix. Since all matrix concatenation is normalized, the selection of 2**14 for the unity element provides an equivalent matrix. See Section 2.3.2.1 for further details of normalization and internal number representations.

ROT

CALL ROT(ANGLE,AXIS)

One of three Matrices is created, dependent upon the axis of rotation:

$$\text{X axis} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & A & B & 0 \\ 0 & -B & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\text{Y axis} = \begin{bmatrix} A & 0 & -B & 0 \\ 0 & I & 0 & 0 \\ B & 0 & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\text{Z axis} = \begin{bmatrix} A & B & 0 & 0 \\ -B & A & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

I = 2**14
A = I*cos(ANGLE)
B = I*sin(ANGLE)

SCALE

CALL SCALE(X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} X & 0 & 0 & 0 \\ 0 & Y & 0 & 0 \\ 0 & 0 & Z & 0 \\ 0 & 0 & 0 & W \end{bmatrix}$$

TRAN

CALL TRAN(X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ X & Y & Z & W \end{bmatrix}$$

WINDOW

CALL WINDOW(WL,WR,WB,WT[,WH,WY[,WE[,W]]])

Two Matrices are produced:
$$\begin{bmatrix} \text{WINDOW} \end{bmatrix} = \begin{bmatrix} W2 \end{bmatrix} \begin{bmatrix} W1 \end{bmatrix}$$

$$W1 = \begin{bmatrix} 1/A & 0 & 0 & 0 \\ 0 & 1/C & 0 & 0 \\ 0 & 0 & 1/E+1/F & 1/F \\ 0 & 0 & 0 & 1/W \end{bmatrix}$$

$$W2 = \begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ -B & -D & -WH & W \end{bmatrix}$$

Where:

A = (WR-WL)/2
B = (WR+WL)/2
C = (WT-WB)/2
D = (WT+WB)/2
E = WY-WH
F = WH-WE
WH = 0 default
WY = W default
WE = minus infinity default

4.2 Multi Picture System Graphics Software Package

The Multi Picture System Graphics Software Package provides twelve subroutines which generate 4x4 matrix transformations. Those subroutines are:

- HBEGIN
- MPINIT
- TIDENT
- TINST
- TMAST
- TROTX
- TROTY
- TROTZ
- TSCALE
- TTRAN
- TWIND
- TWINDP

The use of these subroutines is discussed in the Multi Picture System User's Manual. This section details the 4x4 matrix transformations generated by these subroutines.

HBEGIN

CALL HBEGIN(X,Y,S[,W])

A single Matrix is produced:

$$\begin{bmatrix} W & 0 & 0 & -X \\ 0 & W & 0 & -Y \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & S \end{bmatrix}$$

NOTE: This Matrix is actually the transpose of the hit window matrix. This matrix is concatenated with the current Transformation Matrix in a manner such that both are treated as transposed matrices. This performs a matrix pre-concatenation (rather than post-concatenation as is typically the case). See Section 2.3.2.2, "MATCON" for specific details.

MPINIT

CALL MPINIT

The Transformation Matrix stack is reset and the Transformation Matrix is loaded with:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

$$I = 2^{*}14$$

The value of 2**14 (40000 octal) is chosen for the unity element of the identity matrix rather than 1 to maintain the most precision within the matrix. Since all matrix concatenation is normalized, the selection of 2**14 for the unity element provides an equivalent matrix. See Section 2.3.2.1 for further details of normalization and internal number representations.

TIDENT

CALL TIDENT

A single matrix is produced:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

$$I = 2^{*}14$$

The value of 2**14 (40000 octal) is chosen for the unity element of the identity matrix rather than 1 to maintain the most precision within the matrix. Since all matrix concatenation is normalized, the selection of 2**14 for the unity element provides an equivalent matrix. See Section 2.3.2.1 for further details of normalization and internal number representations.

TINST

CALL TINST(IL,IR,IB,IT,IH,IY[,W])

A single Matrix is produced:

$$\begin{bmatrix} A & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & E & 0 \\ B & D & F & W/2 \end{bmatrix}$$

Where:

$$\begin{aligned} A &= (IR-IL)/4 \\ B &= (IR+IL)/4 \\ C &= (IT-IB)/4 \\ D &= (IT+IB)/4 \\ E &= (IY-IH)/4 \\ F &= (IY+IH)/4 \end{aligned}$$

TMAST

CALL TMAST (ML,MR,MB,MT,MH,MY[,W])

Two Matrices are produced:
$$\begin{bmatrix} \text{TMAST} \end{bmatrix} = \begin{bmatrix} \text{M2} \end{bmatrix} \begin{bmatrix} \text{M1} \end{bmatrix}$$

$$\begin{bmatrix} 1/A & 0 & 0 & 0 \\ 0 & 1/C & 0 & 0 \\ 0 & 0 & 1/E & 0 \\ 0 & 0 & 0 & 2/W \end{bmatrix}$$

$$\begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ -B & -D & -F & W \end{bmatrix}$$

Where:

A = (MR-ML)/4
B = (MR+ML)/4
C = (MT-MB)/4
D = (MT+MB)/4
E = (MY-MH)/4
F = (MH+MY)/4

TROTX

CALL TROTX(ANGLE)

A single matrix is produced:

$$\text{X axis} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & A & B & 0 \\ 0 & -B & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

I = 2**14
A = I*cos(ANGLE)
B = I*sin(ANGLE)

TROTY

CALL TROTY(ANGLE)

A single matrix is produced:

$$\text{Y axis} = \begin{bmatrix} A & 0 & -B & 0 \\ 0 & I & 0 & 0 \\ B & 0 & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

I = 2**14
A = I*cos(ANGLE)
B = I*sin(ANGLE)

TROTZ

CALL TROTZ (ANGLE)

A single matrix is produced:

$$\text{Z axis} = \begin{bmatrix} A & B & 0 & 0 \\ -B & A & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Where:

$I = 2^{*}14$
 $A = I * \cos(\text{ANGLE})$
 $B = I * \sin(\text{ANGLE})$

TSCALE

CALL TSCALE(X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} X & 0 & 0 & 0 \\ 0 & Y & 0 & 0 \\ 0 & 0 & Z & 0 \\ 0 & 0 & 0 & W \end{bmatrix}$$

TTRAN

CALL TTRAN(X,Y,Z[,W])

A single Matrix is produced:

$$\begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ X & Y & Z & W \end{bmatrix}$$

TWIND

CALL TWIND(WL,WR,WB,WT,WH,WY[,W])

Two Matrices are produced:
$$\begin{bmatrix} \text{TWIND} \end{bmatrix} = \begin{bmatrix} \text{W2} \end{bmatrix} \begin{bmatrix} \text{W1} \end{bmatrix}$$

$$W1 = \begin{bmatrix} 1/A & 0 & 0 & 0 \\ 0 & 1/C & 0 & 0 \\ 0 & 0 & 1/E & 0 \\ 0 & 0 & 0 & 1/W \end{bmatrix}$$

$$W2 = \begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ -B & -D & -WH & W \end{bmatrix}$$

Where:

$$\begin{aligned} A &= (WR-WL)/4 \\ B &= (WR+WL)/4 \\ C &= (WT-WB)/4 \\ D &= (WT+WB)/4 \\ E &= (WY-WH)/2 \end{aligned}$$

TWINDP

CALL TWINDP(WL,WR,WB,WT,WH,WY,WE[,W])

Two Matrices are produced: $\begin{bmatrix} & \\ & \end{bmatrix} \text{TWINDP} = \begin{bmatrix} & \\ & \end{bmatrix} \text{W2} \begin{bmatrix} & \\ & \end{bmatrix} \text{W1}$

$$W1 = \begin{bmatrix} 1/A & 0 & 0 & 0 \\ 0 & 1/C & 0 & 0 \\ 0 & 0 & 1/E+1/F & 1/F \\ 0 & 0 & 0 & 1/W \end{bmatrix}$$

$$W2 = \begin{bmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ -B & -D & -WH & W \end{bmatrix}$$

Where:

$$A = (WR-WL)/4$$

$$B = (WR+WL)/4$$

$$C = (WT-WB)/4$$

$$D = (WT+WB)/4$$

$$E = (WY-WH)/2$$

$$F = (WH-WE)/2$$

APPENDIX A

PICTURE SYSTEM SPECIFICATIONS

PICTURE CONTROLLER INTERFACE

	-Connection between Picture System and PDP-11. Requires one Hex Small Peripheral Slot in PDP-11 wired for non-processor request (NPR) and non-processor grant (NPG) operation.
Data Paths	-Direct Input/Output (DIO): Single-word transfer -Direct Memory Access (DMA): Block transfers -Concurrent Operation
Data Transmission	-Up to 500 feet between interface and Picture System. -Differential Driver/Receivers.
Transfer Rate	-DMA to Picture System (not including CPU, UNIBUS, or Memory Access Overhead): (1.0 + .004D) usec/word, where "D" is distance in feet. (1.0 + .0131D) usec/word, where "D" is distance in meters.
Interrupt Types (UNIBUS Address)	-Clock (340) -System (344) -Interactive Devices (350) -DMA (354)
Standard Register Assignments (UNIBUS Address) [Picture Data Bus Address]	-Picture System Data (767660) -DIO Picture System Address (767662) -DMA Word Count (767664) -DMA Bus Address (767666) -I/O Status (767670) -DMA Picture System Address [177747] -DMA Passive Input Port [177770]

PICTURE DATA BUS

	-Connecting link for all Picture System components and devices.
Word Transfer Rate	-150 nsec (6.66 Million wps)
Bus Arbitration	-Synchronous -Round Robin -8 Channels
Bus Word Size	-16 Bits

PICTURE PROCESSOR

URE PROCESSOR

High-speed, micro-programmed, digital arithmetic processor that may be either an Active or Passive device.

Input/Output Registers

-MAP Active Output Limit

[177750]

-MAP Active Output Address

[177751]

-MAP Active Input Address

[177752]

-MAP Status

[177753]

-MAP Passive Output Port

[177776]

-MAP Passive Input Port

[177777]

Micro-Code

-88 Instruction Bits

-256 Instruction words

-150 nsec Cycle Time

Dimension Modes

-2-dimensional data require two 16-bit words to store x and y coordinate values.

-3-dimensional data require three 16-bit words to store x, y and z coordinate values.

-4-dimensional data (homogeneous) require four 16-bit words to store x, y, z and w coordinate values. W serves as a scale factor by which x, y, and z are scaled in order to provide a much larger effective dynamic range.

Data Modes

-ABSOLUTE: specifies coordinates as a displacement from the origin of the data space.

-RELATIVE: specifies coordinates as a displacement from the previous coordinate.

-SET BASE: specifies a coordinate to be loaded into the Base Register.

-OFFSET: specifies coordinates as a displacement from the contents of the Base Register.

-PASS FORMATTED: specifies data to be passed though the MAP unprocessed, but formatted on output for processing by the Line Generator.

-PASS CONDITIONAL: specifies data to be passed through the MAP unprocessed, but output only if the last point processed by the MAP was not clipped. Output data are identical to input data (unformatted).

-PASS: specifies data to be passed through the MAP unprocessed and output unformatted.

PS2/MPS Reference Manual Specifications

Drawing Modes

- MOVE (M): positions the vector point to the specified location.
- DRAW (D): defines a line from the last output location to the specified location.
- The repeat drawing sequences are:
 - * M,D,M,D,...(unconnected lines)
 - * D,M,D,M,...(unconnected lines)
 - * M,D,D,D,...(connected lines)
 - * D,D,D,D,...(connected lines)
 - * M,M,M,M,...(used to process dots)

Output Format Modes

- DISPLAY: transformed, normalized, clipped, perspective divided, and viewport mapped data formatted into two 16-bit words for processing by the Picture Generator.
- SIXTEEN BIT PRECISION: output data formatted into four, 16-bit words for writing fully transformed data back into memory for further processing by the Picture Controller.
- TRANSFORM/NORMALIZE: output data formatted into four 16-bit words which contain the transformed and normalized x, y, z and w coordinates just processed. No windowing, clipping or viewport mapping takes place.
- TRANSFORM ONLY: output data formatted into four 16-bit words which contain the transformed x, y, z and w coordinates just processed. No normalization, windowing, etc. takes place.

Word Length

- 16-bit input.
- 24-bit internal.
- 12 or 16-bit x and y output.
- 6 or 16-bit z output.

Processor Registers

- 256 words of which 208 may be used as a processor matrix stack containing up to 13 4x4 matrices.

Transformations

- TRANSLATION in any direction.
- ROTATION about any axis.
- SCALE with respect to any dimension.
- MIRROR IMAGES about a plane.
- TRANSFORMATIONS expressed as a 4x4 matrix.

PS2/MPS Reference Manual Specifications

Compound Transformations

- Hardware matrix concatenation.
- Transformation matrix may be loaded from or stored into the data base residing in Picture Controller memory.
- Push-down stack for storing 13 full transformation matrices.

Clipping

- Eliminates the portions of objects that are outside the program specified field of view.
- The field of view is a pyramid or frustrum (truncated pyramid) in the data space whose apex is at the eye.
- Clipping is performed with respect to the six program-controlled surfaces of the frustrum.

Perspective

- Displays realistic line representations of three-dimensional objects as they appear to the eye with reference to relative distance or depth.

Instancing

- A method of defining a structure once in the data base and replicating it several times in different positions, sizes and orientations.
- Instancing performed to any level.

Viewport

- The viewport specification is under program control and defines a six surface region of the Picture Display where the picture is to appear. Data which have been transformed, clipped, and put in perspective are linearly mapped into the viewport which allows complete separation of the coordinate systems of the drawing space and the Picture Display.
- The resolution of data mapped into the viewport may be up to 16 bits, which allows data to be used for precision plots.
- Multiple viewports may be defined to give simultaneous use of several areas of the screen.
- Specification of viewport front and back provides the intensity bounds for depth-cueing.

PS2/MPS Reference Manual Specifications

Zooming

-Allows for smooth and rapid motion into (or out of) a data structure in order to obtain a more detailed (or wide angle) view of a chosen region in the drawing space.

Hit Test

-Can detect whether any part of a given picture element is within a program-specified region in the data space. Used for implementing the pointing function. Hit may cause a Picture Controller interrupt.

Processing Speeds (From Picture Memory)

-Including complete MAP processing:

* Matrix Concatenation	31 usec
* Characters (4)	2.4 usec
* Status Word (32 bits)	2.4 usec
* Move or Dot	
Clipped (nonvisible)	8.7 usec
Not Clipped (visible)	11 usec
* Absolute Line Endpoint	
Nonvisible	9.6 usec
Completely Visible	12 usec
Visible, 1 plane	
1 coordinate	16.5 usec
Visible, 1 plane	
2 coordinates	20.6 usec
Visible, 2 planes	
2 coordinates	24.8 usec

Internal Commands

-DRAW: 120 Draw mode combinations.
-HALT: stops the input operation.
-JUMP: transfer of control.
-PUSHJ: transfer of control with current address saved on stack.
-POPJ: return to saved address.
-LOAD: loads the 256 registers.
-STORE: stores the 256 registers.
-LOADSTK: loads data into the stack area of the processor registers.
-STORESTK: stores data from the stack area of the processor registers.
-XFER: transfers contents of the processor registers from one location to another or swaps them.
-PUSH: contents of processor registers loaded onto the stack.
-POP: top of stack loaded into processor registers.
-MATCON: multiplies input matrix by the current transformation matrix.

PS2/MPS Reference Manual
Specifications

PICTURE MEMORY

	-General purpose storage medium for processed and unprocessed data. Its primary purpose is to allow complete separation of display refresh and dynamic update requirements. A passive only device.
Memory Type	-Dual Port. -16 pin, 4Kx1 dynamic MOS IC's.
Word Size	-16 bits.
Memory Size	-16K, 32K, 48K, or 64K words.
Maximum Throughput	-1.33 Millon wps; 1 Active device. -2.22 Millon wps; 2 Active devices (1.11 Millon wps each). -2.50 Millon wps; 3 Active devices (.83 Millon wps each).
Memory Uses	-User Storage. -Untransformed data file. -Transformed display file. -Transformed data file.
Display File Content	-Dots and line endpoint data for use by the Picture Generator. One dot or line endpoint entry requires 2 words of memory: 12 bits for x, 12 bits for y, 6 bits for intensity, and 2 bits for command. -Packed character codes for use by the Character Generator. One entry requires 2 words of memory and contains 4 character codes. -Control information used to direct the operation of the Picture Generator. Each entry requires 2 words of memory.
Display File Size	-Up to 8K entries for each 16K memory module.

PS2/MPS Reference Manual
Specifications

PICTURE GENERATOR

PICTURE SYSTEM 2
REFRESH CONTROLLER

- The Active device in the Picture System 2 responsible for reading processed data from Picture Memory and channelling it to the Line Generator for display on the Picture Display(s).
- Refresh Rate
 - First 16 multiples of 1/120 of a second (1/120, 1/60, 1/40, etc.).
- Buffer Modes
 - Single Buffer
 - Double Buffer
 - Segmented Buffer
- Registers
 - Current Segment Name [177730]
 - Segment Name [177731]
 - Active Writeback Address [177732]
 - Active Writeback Limit [177733]
 - Active Input Address [177734]
 - Active Start Address [177735]
 - Active Input Limit [177736]
 - Status Register [177737]
- Refresh Modes and Timing (per 2 16-bit words)
 - Normal: 1.5 usec.
 - Automatic: 1.5 usec.
 - Memory Compaction: 1.8 usec.
 - Name Search: 1.8 - 2.1 usec.
 - Segment Skip: 1.5 usec.
 - Jump: 3.15 usec.
- Search Modes
 - Next Name
 - Left Byte Match
 - Right Byte Match
 - Word Match

MULTI PICTURE SYSTEM
REFRESH AND DEVICE PROCESSOR

- The Active device in the Multi Picture System responsible for reading processed data from Picture Memory and channelling it to the Line Generator for display on the selected Picture Display(s). Manages Picture Memory so that several tasks can share the Multi Picture System. Also polls the interactive devices, automatically displays Cursor corresponding to Data Tablet stylus position, automatically displays Tracking Cross corresponding to

PS2/MPS Reference Manual
Specifications

Light Pen location and generates interrupts to Picture Controller upon sensing prespecified significant changes in data read from interactive devices.

Refresh Rate

- Automatic refresh initiation; no interrupt to Picture Controller.
- 60 Hz power: 60 Hz refresh, dropping automatically to 40 Hz, 30 Hz, etc., as necessary.
- 50 Hz power: 50 Hz refresh, dropping automatically to 33.3 Hz, 25 Hz, etc., as necessary.

Buffer Modes

- Segmented Buffer
- Double Buffer

Registers

- Refresh Maintenance 0 [177730]
- Refresh Maintenance 1 [177731]
- Refresh Program Counter [177732]
- Refresh Processor Data [177733]
- Refresh Processor Clock [177734]
- Refresh External [177735]

Refresh Modes
and Timing
(per 2 16-bit words)

- Normal: 1.5 usec.
- Memory Compaction: 1.8 usec.

Device Control Modes

- Interrupt on bit change.
- Interrupt on value change.
- Automatic Cursor generation corresponding to Data Tablet stylus position.
- Automatic Tracking Cross generation corresponding to Light Pen movements and "screen blast" to determine Light Pen position.

CHARACTER GENERATOR

Character Font

- Interprets character codes, producing strokes which are sent to the Line Generator for display.

- 128 ASCII Codes.
- 95 displayable ASCII code subsets.
- Fast font allows up to 7000 characters per frame.
- Slow font has more detail for clarity.
- Positioning commands interpreted (i.e., Tab, Carriage Return, Line Feed, Backspace, etc.).
- Programmable left and top margins.
- 256 user programmable codes.
- Programmable inter-character and inter-line spacing.

Character Memory

- 1024 x 12 bit PROM (ASCII Font).
- 1024 x 12 bit RAM (User Programmable Font).

Character Sizes

- 8 sizes standard from .03" to .74".
- 64 user programmable sizes and orientations.
- Italics for all standard character sizes.

Character Orientation

- User Programmable.
- Controlled by 2x2 matrix.
- Superscripts for 6 of the standard character sizes.
- Subscripts for 6 of the standard character sizes.
- ROM storage for 16 sets of character coefficients.
- RAM storage for 32 sets of character coefficients.
- 14-level Font Parameter Stack.

Character Integrity

- No character wrap-around.
- Characters beyond any edge of the screen are not displayed.

PS2/MPS Reference Manual
Specifications

Display Capacity

-For fast font with average character frequency at 30 frames per second refresh rate on 11.23" (28.53 cm) x 11.23" (28.53 cm) display area (60 Hz power source).

<u>Point Size</u>	<u>Height</u>	<u>Characters Per Line</u>	<u>Number of Characters</u>
4	.03" (.082 cm)	342	7000
6	.06" (.13 cm)	228	6200
10	.10" (.21 cm)	147	6700
14	.14" (.29 cm)	98	5600
17	.14" (.35 cm)	81	5300
28	.23" (.58 cm)	49	4300
46	.37" (.95 cm)	30	4300
76	.63" (1.59 cm)	18	4300

-For fast font with average character frequency at 33 frames per second refresh rate on 11.23" (28.53 cm) x 11.23" (28.53 cm) display area (50 Hz power source).

<u>Point Size</u>	<u>Height</u>	<u>Characters Per Line</u>	<u>Number of Characters</u>
4	.03" (.082 cm)	342	7000
6	.06" (.13 cm)	228	6200
10	.10" (.21 cm)	147	6700
14	.14" (.29 cm)	98	5600
17	.14" (.35 cm)	81	5300
28	.23" (.58 cm)	49	4300
46	.37" (.95 cm)	30	4300
76	.63" (1.59 cm)	18	4300

PS2/MPS Reference Manual
Specifications

LINE GENERATOR

Line Modes

-Converts digital end-point coordinates to analog signals for drawing on the Picture Display. The following specifications pertain to the standard monochrome display.

-Blink
-Short dashed - - - - -
-Medium-short - - - - -
-Medium-long - - - - -
-Long - - - - -
-Long-short - - - - -
-Long-short-short - - - - -
-Solid - - - - -
-Endpoints drawn as dots.
-Continuous texture mode for dashed lines.

Display Select

-Up to 6 Picture Displays. For the Multi Picture System, the 6 displays may be arranged in a maximum of 4 Picture Stations.

Intensity

-64 levels of constant intensity.
-Depth-cueing.

Drawing Speeds
(14"x14" viewing area)

-Line length (L) < .155" = 1.55 usec.
-.155" < L < .218" = (1.15 + 2.586L) usec.
-L > .218" = (1.6 + 1.293L) usec.

Drawing Speeds For
System Using Light Pen
(14"x14" viewing area)

-Line length (L) < .155" = 1.75 usec.
-.155" < L < .218" = (1.6 + 2.586L) usec.
-L > .218" = (1.6 + 1.293L) usec.

Move Speed
(14"x14" viewing area)

-Move length (M) < .696" = 1.55 usec.
-M > .696" = (1.1 + .647M) usec.

Dot Speed
(14"x14" viewing area)

-Dot spacing (D) < .386" = 1.55 usec.
-D > .386" = (1.3 + .647D) usec.

PS2/MPS Reference Manual
Specifications

Display Capacity

-At 30 frames per second refresh rate
for 60 Hz power:

<u>Length</u>	<u>Moves</u>	<u>Dots</u>	<u>Connected Lines</u>
.1"	21,500	21,500	21,500
.2"	21,500	21,500	20,000
.5"	21,500	20,530	14,830
1"	19,080	17,120	11,520
2"	13,920	12,850	7,960
5"	7,690	7,350	4,130
6"	6,690	6,430	3,560
10"	4,400	4,290	2,290
12"	3,760	3,675	1,940
14"	3,280	3,215	1,690

-At 33 frames per second refresh rate
for 50 Hz power:

<u>Length</u>	<u>Moves</u>	<u>Dots</u>	<u>Connected Lines</u>
0.25 cm	19,350	19,350	19,350
0.50 cm	19,350	19,350	17,990
1.27 cm	19,350	18,470	13,350
2.54 cm	17,170	15,400	10,360
5.08 cm	12,530	11,560	7,160
12.70 cm	6,920	6,610	3,710
15.24 cm	6,020	5,780	3,200
25.40 cm	3,960	3,860	2,060
30.48 cm	3,380	3,300	1,750
35.56 cm	2,950	2,890	1,520

PS2/MPS Reference Manual
Specifications

PICTURE DISPLAY

Display Type	-Cathode Ray Tube (CRT).
Deflection Type	-Calligraphic (random stroke).
Addressable Locations	-Electromagnetic with phosphor protection, overdrive protection and dynamic focus.
Display Screen Size	-4096(X) x 4096(Y) x 64(intensity).
Phosphor	-21" (53.34 cm) rectangular. -16" (40.64 cm) horizontal x 13" (33 cm) vertical opening. -10" (25.4 cm) x 10" (25.4 cm) quality viewing area.
Controls	-P4 standard. -Others available.
Spot Size	-On/Off -Focus -X and Y Gain -X and Y Position -Intensity -Brightness -Contrast (5 selectable values)
Endpoint Match	-Not exceeding 0.020" (0.0508 cm).
Brightness	-Less than 1/2 line width.
Jitter	-10 foot-lamberts with 10" (25.4 cm) x 10" (25.4 cm) raster of 512 lines at 40 Hz frame with P4 phosphor within specified spot size.
Contrast	-.005" (0.0127 cm) maximum.
Deflection Input Impedance	-4:1 or greater.
Deflection Sensitivity	-Greater than 1K ohm.
Video Input Impedance	-1 inch per volt.
Video Sensitivity	-500 ohms.
Unblank Input Impedance	-0 to 2.5 volts (2.5 volts = maximum intensity).
Unblank Sensitivity	-0 to +2.5 Volts (2.5 volts unblanked).

GRAPHICS PERIPHERALS

ALPHANUMERIC KEYBOARD

Physical Description

-Keys struck may be interpreted and acted upon by the user program.

- 61 keys.
- Shift/Unshift keys.
- Control keys.
- Enclosure dimensions:
 - 3.50" (8.9 cm) height (maximum).
 - 16.25" (41.3 cm) width.
 - 8.00" (20.3 cm) depth.
- Enclosure designed to fit into optional table cut-out on Light Pen Picture Station.

Character Codes

-Standard 128 ASCII character set.

Interface

- Picture Data Bus interface.
- In conjunction with the Picture Display, may optionally function as a general alphanumeric terminal. See Remote Terminal Interface option for further details.

CONTROL DIALS

Physical Description

-Dial positions are encoded into information which may be interpreted and acted upon by the user program.

- 8 knobs (continuous-turn potentiometers) per Control Dials unit.
- Enclosure dimensions for standard model:
 - 3.50" (8.9 cm) height (maximum).
 - 14.00" (35.6 cm) width.
 - 5.75" (14.6 cm) depth.
- Enclosure dimensions for special Picture Station model designed to fit into a tray with the Lighted Function Buttons:
 - 6.00" (15.2 cm) height (maximum).
 - 4.25" (10.8 cm) width.
 - 7.85" (19.9 cm) depth.

Interface

- Picture Data Bus interface.
- 16 channel 12-bit analog-to-digital converter.
- One interface accommodates two Control Dials units or one Control Dials unit and two Joysticks.

PS2/MPS Reference Manual
Specifications

DATA TABLET

	-General purpose interactive graphics input device, inherently well-suited to entry of precise positional information.
Physical Description	-Tablet Surface (standard model): 0.75" (1.9 cm) height. 19.00" (48.3 cm) width. 19.00" (48.3 cm) depth. -Tablet electronics cabinet: 7.25" (18.4 cm) height. 9.50" (24.1 cm) width. 11.25" (28.6 cm) depth. -Stylus with 2-position tip switch.
Output	-11 bits of X. -11 bits of Y. -Pen proximity and up/down status.
Resolution	-200 lines per inch.
Proximity Range	-3/16" (0.48 cm).
Repetition Rate	-Continuously variable up to 200 coordinate pairs per second.
Active Area	-11" x 11" (27.9 cm x 27.9 cm) standard. -20" x 20" (50.8 cm x 50.8 cm), 36" x 48" (91.4 cm x 121.9 cm) and other sizes are optionally available.
Interface	-Picture Data Bus interface.

FUNCTION SWITCHES & LIGHTS

Physical Description

- Settings of 16 switches and 16 independently programmable lights are encoded into information which may be interpreted and acted upon by the user program.
- 16 3-position switches:
 - Up position is locked on.
 - Centered position is neutral (off).
 - Down position is momentary on with spring return to center.
- 16 independently program controlled lights.
- Overlay sheets and cover provided.
- Enclosure dimensions:
 - 3.50" (8.9 cm) height (maximum).
 - 14.00" (35.6 cm) width.
 - 5.75" (14.6 cm) depth.

Interface

- Picture Data Bus interface.

JOYSTICK

Physical Description

- A general purpose graphic input device for positioning and tracking functions.
- Single control unit with 3-axis input potentiometers.
- Spring return to neutral on all axis movement.
- Each axis has individual adjustment for neutral position.
- Enclosure dimensions:
 - 5.75" (15.6 cm) height (maximum).
 - 7.25" (18.4 cm) width.
 - 5.80" (14.7 cm) depth.

Interface

- Picture Data Bus interface.
- 16 channel 12-bit analog-to-digital converter.
- One interface accommodates four Joystick units or two Joystick units and one Control Dials unit.

PS2/MPS Reference Manual
Specifications

LIGHT PEN

- Physical Description
- A device available only with Multi Picture System for user interaction which allows the selection of dots, lines and characters from the data being displayed on the Picture Display.
- Acceptance Area
- 0.5" (1.27 cm) diameter cylinder which is 6" (15.2 cm) in length.
 - Two position tip switch.
 - Coiled cord with 48" (121.9 cm) maximum length and 14" (35.6 cm) minimum length.
- Response Time
- 0.075" (.19 cm) diameter.
- Interface
- Less than 300 nsec.
 - Picture Data Bus interface.
 - One interface accommodates two Light Pens.

LIGHTED FUNCTION BUTTONS

- Physical Description
- Settings of 32 lightable buttons are encoded into information which may be interpreted and acted upon by the user program.
 - 32 independently programmable push buttons.
 - Associated program controlled lights underneath each button.
 - Pegs for locating overlay sheets.
 - Enclosure designed to fit into tray with special Picture Station model Control Dials.
 - Enclosure dimensions:
 - 6.50" (16.5 cm) height (maximum).
 - 6.50" (16.5 cm) width.
 - 7.85" (19.9 cm) depth.
- Interface
- Picture Data Bus interface.
 - One interface will accommodate two Lighted Function Buttons units.

REMOTE TERMINAL INTERFACE

- Connection between the host computer and the Picture System 's Picture Display and Alphanumeric Keyboard that enable the display and keyboard to function as a standard alphanumeric terminal.
- Operates completely independently of the Picture System Picture Processor, Picture Memory and Refresh Controller (Picture System 2) or Refresh and Device Processor (Multi Picture System).
- Shares the Picture Generator's Line Generator and Character Generator components to display the characters on the Picture Display.
- Internal character memory, distinct from the Picture Memory, with sufficient storage for 2048 characters.

Interface

- EIA or 20 ma. current loop connection to host computer interface.

Prerequisites

- EIA or 20 ma. current loop interface in host computer and modem or null modem.

PICTURE SYSTEM CABINETRY

- The Picture System may be configured in a variety of ways depending on hardware configuration and individual customer requirements. The cabinetry consists of the electronics cabinet, which contains all of the hardware components of the Picture System, and the console work station(s), or Picture Station(s). Each Picture Station consists of one or more work tables, one or more Picture Displays and optionally a Picture Station chair.
- Modular construction.
- All units furnished with black laminated plastic surface with rounded corners, edged by soft black plastic bumper.
- All tables supported by round steel legs, 3" (7.62 cm) diameter with adjustable levelers, finished in black enamel.

ELECTRONICS CABINET

- 29.5" (75 cm) height.
- 27.5" (70 cm) width.
- 36.5" (93 cm) depth.
- Removable black plastic side panels which allow access to all electronic modules.
- Removable work surface.
- Provided with casters for easy movement and adjustable feet to lock cabinet into stationary position.

STANDARD WORK TABLE

- 29.5" (75 cm) height.
- 44.5" (113 cm) width.
- 30.0" (76 cm) depth.

LIGHT PEN WORK TABLE

- 29.50" (75 cmm) height.
- 42.25" (107 cm) width.
- 36.25 (92 cm) depth.
- Equipped with stand-off mountings which hold the Picture Display at a convenient position for Light Pen interaction and permit tilting of display screen. Tilt lock knobs firmly hold display at angle desired.
- Reinforced with steel baffle.
- Furnished with or without Alphanumeric Keyboard cutout.

PS2/MPS Reference Manual
Specifications

LONG SIDE-TABLE

- 29.50" (75 cm) height.
- 73.00" (185 cm) width.
- 27.25" (69 cm) depth.

SHORT SIDE-TABLE

- 29.50" (75 cm) height.
- 36.50" (93 cm) width.
- 27.25" (69 cm) depth.

PICTURE STATION CHAIR

- Human engineered to reduce fatigue of operator during extended periods of operation.
- Black enamelled steel frame.
- Adjustable height and tilt.
- Caster supported legs on swivel base.
- Black leather or durable nylon fabric covered seat and back cushions (specify when ordering).

DISPLAY PEDESTAL

- Companion to standard work table for Picture Display support.
- Five legs, castor supported with wheel locks for solid support on even or uneven surfaces.
- Permits swivel and tilt adjustments of Picture Display. Tilt lock knobs included.

TRAY

- Encloses Lighted Function Buttons unit and special workstation model Control Dials unit.
- Dimensions:
 - 10.8" (27.4 cm) width.
 - 7.9" (20.1 cm) depth.

CABLES

- Interface cable 25 feet (7.62 m) in length between host computer and electronics cabinet; optional lengths are 50 feet (15.24 m), 75 feet (22.86 m), 100 feet (30.48 m), 150 feet (45.72 m), 300 feet (91.44 m) and 500 feet (152.40 m).
- Cable stubs at each end of interface cable included.

PS2/MPS Reference Manual Specifications

- Remote power cable between host computer and electronics cabinet required for controlling Picture System power on/off switch; same lengths as interface cable.
- Cable between electronics cabinet and Picture Display 15 feet (4.57 m) in length of which about 7.5 feet (2.29 m) is useable; optionally up to 25 feet (7.62 m) of which about 17.5 feet (5.33 m) is useable.
- All Graphics Peripherals are supplied with 12 feet (3.66 m) of cable of which 7 feet (2.13 m) is useable.
- Remote Terminal Interface cable 25 feet (7.62 m) in length between electronics cabinet and modem or null modem that serves as communications link to host computer; optional lengths are 50 feet (15.24 m), 75 feet (22.86 m) and 100 feet (30.48 m).

POWER REQUIREMENTS

- 107 to 127 volts, 60 Hz.
 - Single Phase.
 - 30 Amp.
 - Two wire plus ground.
 - Ground common with host computer in the Picture System interface resides.
 - Hubbel 2610 or equivalent.
 - 2160 Watts.
 - 7400 BTU per hour.
- | | |
|-------------------|--|
| Receptacle | |
| Power Consumption | |
| Heat Dissipation | |

ENVIRONMENTAL REQUIREMENTS

- | | |
|-----------------------|-----------------------------------|
| Operating Temperature | -65 F to 80 F (18 C to 27 C). |
| Relative Humidity | -20% to 80%. |
| Recommended Clearance | -3 feet (about 1 m) on all sides. |

APPENDIX B

PROGRAMMING THE PICTURE SYSTEM

B.1 Introduction

Chapter 2 of this manual described in detail the Picture System/PDP-11 Interface and Picture System registers used in programming the system. The purpose of this appendix is to show how these registers may be used in a program which interfaces with the hardware at the assembly language level.

B.2 Program Description

To illustrate how to program the Picture System, a simple program will be detailed which displays a cube and allows it to be translated in the x, y and z directions according to console switch settings. During this program, the characters "CUBE" will be displayed blinking continuously.

In addition to the program details, the following points should be emphasized:

1. During initialization, the various devices must be set to the configuration as it is intended to be used. For this program, the system is configured in the following manner:
 - a. DMA (Active Output) => MAP Input Controller (Passive Input).
 - b. MAP Output Formatter (Active Output) => PS Memory (Passive Input).
 - c. PS Memory (Passive Output) => Refresh Controller (Active).
 - d. Refresh Controller (Active) => Line Generator (Passive Input).
2. The number of data words transferred must correspond for the RSR Command that is being executed by the Picture Processor in order to ensure the operation continues to completion. Exactly what the correspondence is depends upon the RSR command specified. For the

PS2/MPS Reference Manual
Example Program

2DDRAW, 3DDRAW and 4DDRAW RSR commands, the CNT field of the RSR will contain the two's complement (sign implied) of the number of executions to be performed and the DMAWC will contain the two's complement of the number of 16-bit words to be transferred. The following illustrates this relationship:

2DDRAW: CNT $\leq$ -(number of 16-bit word pairs)
DMAWC $\leq$ 2*CNT

3DDRAW: CNT $\leq$ -(number of 16-bit word triples)
DMAWC $\leq$ 3*CNT

4DDRAW: CNT $\leq$ -(number of 16-bit 4-word sets)
DMAWC $\leq$ 4*CNT

3. Before a 2DDRAW or 3DDRAW RSR command is performed (except for PASS commands, FSM2=5-7), the BASE register should be loaded with the constant Z and W coordinates for a 2DDRAW or the constant W coordinate for a 3DDRAW. The BASE register is loaded by a 4DDRAW Set Base (FSM2=0) RSR command. This is done since the BASE register supplies the Z and W coordinates for 2DDRAW commands and the W coordinate for the 3DDRAW command.
4. It should be noted that the Status Command (see Section 2.4.3) is deposited directly into the refresh buffer. Once a Status Command is encountered by the Picture Generator, the status specified will remain in effect (through subsequent frames when no Status Command is encountered) until an overriding Status Command is encountered.

The structure of this sample program is shown in Figure B-1. The following section contains the MACRO-11 assembly language listing of the program. Careful study of the listing should clarify many of the topics covered within this manual.

Figure B-1
Example Program Structure

PS2/MPS Reference Manual
Example Program

B.3 MACRO-11 Picture System 2 Programming Example

APPENDIX A: SAMPLE PROGRAM
TABLE OF CONTENTS

MACRO M11

11-JAN-77 08:57

2-	28	MACROS
3-	80	DEFINITION OF REGISTERS AND COMMANDS
4-	130	DEFINITION OF CONSTANTS AND DATA
5-	209	PROGRAM INITIALIZATION
6-	293	RTC INTERRUPT SERVICE ROUTINE
7-	347	DATA TO INPUT?
9-	395	PICTURE DISPLAY
10-	500	DMA OUTPUT ROUTINE

```

*TITLE      APPENDIX A: SAMPLE PROGRAM
*
*EVANS & SUTHERLAND COMPUTER CORPORATION      COPYRIGHT (C) 1976 ;
*
*THIS SAMPLE PROGRAM DISPLAYS A CUBE IN PERSPECTIVE WITHIN A
*VIEWPORT WHICH IS SPECIFIED AS THE ENTIRE SCREEN USING PDP-11
*CODE WHICH INTERFACES DIRECTLY WITH PICTURE SYSTEM 2. THE
*CUBE MAY BE TRANSLATED IN X, Y AND Z DIRECTIONS DEPENDENT UPON
*THE CONSOLE DATA SWITCH SETTINGS, ALL THE WHILE DISPLAYING THE
*CHARACTERS "THIS IS A CUBE" WHICH BLINK CONTINUALLY.
*
*CONSOLE SWITCH SETTINGS:
*
*SWITCH 15:  PROGRAM RESTART
*SWITCH 14:  TRANSLATE -X
*SWITCH 13:  TRANSLATE -Y
*SWITCH 12:  TRANSLATE -Z
*SWITCH 11:  TRANSLATE +X
*SWITCH 10:  TRANSLATE +Y
*SWITCH 9:   TRANSLATE +Z
*
*M. MANTLE 11/22/76

```

```
28      .SBTIL  MACROS
29
30      ; THESE MACROS ARE IMPLEMENTED TO ALLOW THE MANIPULATION OF
31      ; PICTURE SYSTEM 2 MEMORY AND SCB LOCATIONS AS IF THEY WERE
32      ; DIRECTLY ADDRESSABLE FROM THE PDP-11.
33
34      .MACRO  RDPS      SRC,DST      ;READ PS2 "SRC" TO PDP-11 "DST"
35
36          TST          IOST
37          BPL          .-4
38          MOV          SRC,DIOPSA
39          TST          IOST
40          BPL          .-4
41          MOV          PSDATA,DST
42
43      .ENDM  RDPS
44
45      .MACRO  WTPS      SRC,DST      ;WRITE PDP-11 "SRC" TO PS2 "DST"
46
47          TST          IOST
48          BPL          .-4
49          MOV          DST,DIOPSA
50          TST          IOST
51          BPL          .-4
52          MOV          SRC,PSDATA
53
54      .ENDM  WTPS
55
56      .MACRO  RWPS      INST,SRC,DST ;PERFORM PDP-11 "INST"
57                                     ;FROM PDP-11 "SRC" TO PS2 "DST"
58                                     ;OR ON PS2 "SRC" IF ONE OPERAND
59
60
61          TST          IOST
62          BPL          .-4
63
64      .IF      B          DST
65          MOV          SRC,DIOPSA
66          TST          IOST
67          BPL          .-4
68          INST        PSDATA
69
70      .IFF
71          MOV          DST,DIOPSA
72          TST          IOST
73          BPL          .-4
74          INST        SRC,PSDATA
75
76      .ENDC
77      .ENDM  RWPS
78
```

.SBTTL DEFINITION OF REGISTERS AND COMMANDS

80			
81			
82	000000	R0	=%0
83	000001	R1	=%1
84	000002	R2	=%2
85	000003	R3	=%3
86	000004	R4	=%4
87	000005	R5	=%5
88	000006	SP	=%6
89	000007	PC	=%7
90			
91	000340	LOWVEC	=000340 ;LOWEST PS2 INTERRUPT VECTOR
92			
93	167660	PSDATA	=167660 ;PS2 PSDATA UNIBUS REGISTER
94	167662	DIOPSA	=PSDATA+2 ;PS2 DIOPSA UNIBUS REGISTER
95	167664	DMAWC	=PSDATA+4 ;PS2 DMAWC UNIBUS REGISTER
96	167666	DMABA	=PSDATA+6 ;PS2 DMABA UNIBUS REGISTER
97	167670	IOST	=PSDATA+10 ;PS2 IOST UNIBUS REGISTER
98	177570	SWITCH	=177570 ;CONSOLE SWITCH REGISTER
99			
100	177735	RFASA	=177735 ;PS2 RFASA SCB REGISTER
101	177736	RFAIL	=RFASA+1 ;PS2 RFAIL SCB REGISTER
102	177737	RFSR	=RFASA+2 ;PS2 RFSR SCB REGISTER
103			
104	177744	RTCCNT	=177744 ;PS2 RTCCNT SCB REGISTER
105	177745	RTCSR	=RTCCNT+1 ;PS2 RTCSR SCB REGISTER
106			
107	177747	DMAPSA	=177747 ;PS2 DMAPSA SCB REGISTER
108			
109	177750	MAOL	=177750 ;PS2 MAOL SCB REGISTER
110	177751	MAOA	=MAOL+1 ;PS2 MAOA SCB REGISTER
111	177752	MAIA	=MAOL+2 ;PS2 MAIA SCB REGISTER
112	177753	MSR	=MAOL+3 ;PS2 MSR SCB REGISTER
113	177754	MMSR	=MAOL+4 ;PS2 MMSR SCB REGISTER
114			
115	177760	RTCREQ	=177760 ;PS2 RTCREQ SCB REGISTER
116	177761	RTCIE	=RTCREQ+1 ;PS2 RTCIE SCB REGISTER
117			
118	177775	LGPIP	=177775 ;PS2 LGPIP SCB REGISTER
119	177777	MPIP	=177777 ;PS2 MPIP SCB REGISTER
120			
121	010000	LOAD	=010000 ;LOAD RSR COMMAND
122	022360	PUSH	=022360 ;PUSH RSR COMMAND
123	024360	POP	=024360 ;POP RSR COMMAND
124	026000	MATCON	=026000 ;MATCON RSR COMMAND
125	040000	DRAW2D	=040000 ;BASIC 2DDRAW RSR COMMAND
126	100000	DRAW3D	=100000 ;BASIC 3DDRAW RSR COMMAND
127	140377	SETBAS	=140377 ;4DDRAW (SETBASE) RSR COMMAND
128			

```
.SBTIL  DEFINITION OF CONSTANTS AND DATA

130
131
132 000000 BEGIN:
133 000000 000000 MAPREG: .WORD 0 ;BEGIN LOADING MAP REGISTER 0
134 000002 000000 000000 000000 .WORD 0,0,0,77777 ;0-3: INPUT/BASE
      000010 077777
135 000012 000000 000000 000000 .WORD 0,0,0,77777 ;4-7: BASE/INPUT
      000020 077777
136 000022 000000 000000 000000 .WORD 0,0,0,0 ;10-13: MOUT/SAVE
      000030 000000
137 000032 000000 000000 000000 .WORD 0,0,0,0 ;14-17: SAVE/MOUT
      000040 000000
138 000042 003777 .WORD 2047. ;20: VIEWXH
139 000044 000000 .WORD 0 ;21: VIEWXC
140 000046 003777 .WORD 2047. ;22: VIEWYH
141 000050 000000 .WORD 0 ;23: VIEWYC
142 000052 177701 .WORD -63. ;24: VIEWZS (MAX DEPTH CUEING)
143 000054 000077 .WORD 63. ;25: VIEWZF
144 000056 000000 .WORD 0 ;26: NOT USED
145 000060 000057 .WORD 57 ;27: TMADR
146 000062 000000 000000 000000 .WORD 0,0,0,0 ;30-33: NEWCLIP/CLIPSAVE
      000070 000000
147 000072 000000 000000 000000 .WORD 0,0,0,0 ;34-37: CLIPSAVE/NEWCLIP
      000100 000000
148
149 000102 040000 000000 000000 .WORD 40000,0,0,0 ;40-57: INDENTITY MATRIX
      000110 000000 ;(DIAGONAL=40000)
150 000112 000000 040000 000000 .WORD 0,40000,0,0
      000120 000000
151 000122 000000 000000 040000 .WORD 0,0,40000,0
      000130 000000
152 000132 000000 000000 000000 .WORD 0,0,0,40000
      000140 040000
153
154 ; INITIAL PICTURE GENERATOR STATUS
155
156 000142 000200 176000 PGINIT: .WORD 200,176000 ;INITIAL LG STATUS (ALL SCOPES)
157 000146 003 000 005 .BYTE 3,0,5,0 ;LOAD CG FONT PARAMETER STACK
      000151 000
158
159 ; BLINK STATUS
160
161 000152 020200 176000 BLNKON: .WORD 20200,176000 ;SELECT BLINK AND ALL SCOPES
162 000156 000200 176000 BLNKOF: .WORD 200,176000 ;SELECT ONLY ALL SCOPES
163
164 ; PERSPECTIVE MATRIX
165
166 000162 040000 000000 000000 PERMAT: .WORD 40000,0,0,0 ;40000,00000,00000,00000
      000170 000000
167 000172 000000 040000 000000 .WORD 0,40000,0,0 ;00000,40000,00000,00000
      000200 000000
168 000202 000000 000000 040000 .WORD 0,0,40000,20000 ;00000,00000,40000,20000
      000210 020000
169 000212 000000 000000 000000 .WORD 0,0,0,40000 ;00000,00000,00000,40000
      000220 040000
170
171
```



```

172                                     ; TRANSLATION MATRIX
173
174 000222 040000 000000 000000 TRNMAT: .WORD 40000,0,0,0 ;40000,00000,00000,00000
    000230 000000
175 000232 000000 040000 000000 .WORD 0,40000,0,0 ;00000,40000,00000,00000
    000240 000000
176 000242 000000 000000 040000 .WORD 0,0,40000,0 ;00000,00000,40000,00000
    000250 000000
177 000252 000000 TX: .WORD 0 ; TX, TY, TZ,40000
178 000254 000000 TY: .WORD 0
179 000256 000000 040000 TZ: .WORD 0,40000
180
181                                     ; BASE REGISTER VALUES
182
183 000262 000000 000000 000000 ZWBASE: .WORD 0,0,0,77777 ;Z=0, W=77777 (W=32767.)
    000270 077777
184
185                                     ; VARIABLES AND DATA
186
187 000272 154360 154360 SEIPI: .WORD -10000,-10000.
188 000276 124 110 TEXT: .ASCII 'THIS IS A CUBE'
    000301 123 040 111
    000304 123 040 101
    000307 040 103 125
    000312 102 105
    000314 000 000
189 .BYTE 0,0 ;(CHARACTERS MUST BE OUTPUT
190 ; IN MULTIPLES OF 4)
191
192 000316 004000 004000 174000 CDAT1: .WORD 2048.,2048.,-2048., -2048.,2048.,-2048.
    000324 174000 004000 174000
193 000332 174000 174000 174000 .WORD -2048.,-2048.,-2048., 2048.,-2048.,-2048.
    000340 004000 174000 174000
194 000346 004000 004000 174000 .WORD 2048.,2048.,-2048., 2048.,2048.,2048.
    000354 004000 004000 004000
195 000362 174000 004000 004000 .WORD -2048.,2048.,2048., -2048.,-2048.,2048.
    000370 174000 174000 004000
196 000376 004000 174000 004000 .WORD 2048.,-2048.,2048., 2048.,2048.,2048.
    000404 004000 004000 004000
197 000412 174000 004000 174000 CDAT2: .WORD -2048.,2048.,-2048., -2048.,2048.,2048.
    000420 174000 004000 004000
198 000426 174000 174000 174000 .WORD -2048.,-2048.,-2048., -2048.,-2048.,2048.
    000434 174000 174000 004000
199 000442 004000 174000 174000 .WORD 2048.,-2048.,-2048., 2048.,-2048.,2048.
    000450 004000 174000 004000
200 000456 000000
201 000460 000310
202 000462 177776
203
204 000464 000000
205 000466 000000 000000 000000 RBTB: .WORD 0,0,0 ;DO NO SEPARATE
206
207

```

```
209 .SBTTL PROGRAM INITIALIZATION
210
211 000474 012706 000000' START: MOV #BEGIN,SP ;SET THE STACK POINTER
212
213 ; RESET PICTURE SYSTEM 2
214
215 000500 052767 020000 167670 BIS #20000,IOST ;ISSUE A PSRESET
216 ;THIS DOES THE FOLLOWING:
217 ; DIOPSA <= 0
218 ; DMAWC <= 0
219 ; DMABA <= 0
220 ; IOST <= 100200
221 ; DMAPSA <= 0
222
223 ; INITIALIZE THE DMA
224
225 000506 WTPS #MPIP,#DMAPSA ;SET THE DMA TO ADDRESS THE MPIP
226
227 ; INITIALIZE THE PICTURE PROCESSOR
228
229 ;MAP INPUT CONTROLLER WAS SET
230 ;PASSIVE BY PSRESET
231
232 000536 RWPS BIC #6,#MMSR ;ENABLE MAP-CLEAR MAPMNT & MAPHLT
233 ;MAP INTERNAL REGISTERS (0-57)
234 000566 012700 010320 MOV #LOAD+320,R0 ;SET LOAD RSR
235 000572 012701 177717 MOV #-61,R1 ;SET TO TRANSFER 61 (OCTAL) WORDS
236 000576 012702 000000' MOV #MAPREG,R2 ;SET BASE ADDRESS FOR TRANSFER
237 000602 004767 001554 JSR PC,DMAOUT ;AND DMA OUT THE DATA
238
239 ;MAP OUTPUT FORMATTER
240 000606 RWPS CLR #MAOA ;SET PS2 REFRESH BUFFER LIMITS
241 000634 WTPS #8190.,#MAOL ;ASSUME 16K MEMORY (DOUBLE BUFFER)
242 000664 RWPS BIS #40,#MSR ;SET MAP OUTPUT FORMATTER ACTIVE
243
244 ; INITIALIZE PSMEM LIMITS FOR 16K REFRESH BUFFER
245
246 000714 005067 177544 CLR RBTB-2 ;SET BUFFER 1 LOW START ADDRESS
247 000720 012767 017776 177540 MOV #8190.,RBTB ;SET BUFFER 1 HIGH LIMIT ADDRESS
248 000726 012767 020000 177534 MOV #8192.,RBTB+2 ;SET BUFFER 2 LOW START ADDRESS
249 000734 012767 037776 177530 MOV #16382.,RBTB+4 ;SET BUFFER 2 HIGH LIMIT ADDRESS
250 000742 012767 000002 177512 MOV #2,BUFPTR ;INITIALIZE THE BUFFER POINTER
251 ;BUFPTR & RBTB USED BY RTC ISR
252 000750 WTPS #8192.,#RFASA ;SET INITIAL REFRESH LIMITS
253 001000 WTPS #8192.,#RFAIL
254
255 ; INITIALIZE PICTURE GENERATOR STATUS
256
257 001030 012701 000004 MOV #4,R1 ;SET # OF WORDS TO TRANSFER
258 001034 012702 000142' MOV #PGINIT,R2 ;SET BASE ADDRESS
259 001040 1$: WTPS (R2)+,#LGPIP ;WRITE DIRECTLY TO LGPIP
260 001066 005301 DEC R1 ;DONE?
261 001070 003363 BGT 1$ ;IF GT NO
262
263 ; INITIALIZE INTERRUPT VECTORS, INTERRUPT ENABLES & DEVICE STATUS
264
265 001072 012700 000340 MOV #LOWVEC,R0 ;SET LOW VECTOR ADDRESS
```

APPENDIX A: SAMPLE PROGRAM
PROGRAM INITIALIZATION

MACRO M11

11-JAN-77 08:57 PAGE 5-1

```

266 001076 012720 001300'      MOV      #RTCINT,(R0)+      ;SET RTC ISR ADDRESS
267 001102 012720 000240'      MOV      #240,(R0)+      ;AND PSW
268 001106 012720 001726'      MOV      #NOINT,(R0)+      ;SET SYSTEM ISR ADDRESS
269 001112 012720 000240'      MOV      #240,(R0)+      ;AND PSW
270 001116 012720 001726'      MOV      #NOINT,(R0)+      ;SET DEVICE ISR ADDRESS
271 001122 012720 000240'      MOV      #240,(R0)+      ;AND PSW
272 001126 012720 001726'      MOV      #NOINT,(R0)+      ;SET DMA ISR ADDRESS
273 001132 012720 000240'      MOV      #240,(R0)+      ;AND PSW
274
275 001136                      WTPS      #1,#RTICIE      ;ENABLE THE RTC INTERRUPT
276 001166                      WTPS      #16,#RTCCNT      ;SET CLOCK COUNT (60 HZ.)
277 001216                      BIS       #1,#RTCSR      ;AND START THE RTC
278
279 001246 052767 000400 167670  BIS       #400,I0ST      ;SET PSIE AND DMAIE
280
281      ; INITIALIZE VARIABLES
282
283 001254 005067 177176          CLR       NUFRAM
284 001260 005067 176766          CLR       TX
285 001264 005067 176764          CLR       IY
286 001270 005067 176762          CLR       TZ
287
288      ; AND OUTPUT AT LEAST THE FIRST FRAME
289
290 001274 000167 000560          JMP       DSPLAY      ;BRANCH AROUND THE DATA INPUT
291

```

```

293      .SBTTL  RTC INTERRUPT SERVICE ROUTINE
294
295      RTCINT:  MOV     R5,-(SP)      ;SAVE R5
296      MOV     R4,-(SP)      ;SAVE R4
297      1$:      TST     IOST        ;DIO READY?
298      BPL     1$            ;IF PL NO...WAIT FOR IT
299      TST     DIOPSA        ;READ THE CURRENT DIOPSA
300      2$:      TST     IOST        ;DIO READY?
301      BPL     2$            ;IF PL NO...WAIT FOR IT
302      MOV     PSDATA,-(SP)      ;SAVE THE CURRENT DIOPSA
303
304      RWPS      TST     #RFSR        ;RFSTOPPED?
305      BPL     5$            ;IF PL NO...JUST EXIT
306
307      TST     NUFRAM        ;IS A NEW FRAME READY?
308      BEQ     4$            ;IF EQ NO...JUST RESTART REFRESH
309
310      310 001366 032767 000200 167670      BIT     #200,IOST      ;DMA DONE?
311      311 001374 001515          ,          BEQ     4$            ;IF EQ NO...JUST RESTART REFRESH
312
313      313 001376          ,          3$:      TST     #MSR          ;PPDONE?
314      314 001376          ,          RWPS      BPL     3$            ;IF PL NO...LOOP
315      315 001424 100364          ,          RDPS     #MAOA,R4      ;GET CURRENT MAOA
316      316 001426          ,          MOV     BUFPIR,R5      ;GET CURRENT BUFFER POINTER
317      317 001454 016705 177002          ,          WTPS     RBTB(R5),#MAOA      ;SET RB LIMITS FOR NEXT FRAME
318      318 001460          ,          WTPS     RBTB+2(R5),#MAOL
319      319 001510          ,          NEG     R5              ;TOGGLE BUFFER POINTER
320      320 001540 005405          ,          MOV     R5,BUFPIR      ;SAVE BUFFER POINTER
321      321 001542 010567 176714          ,          WTPS     RBTB(R5),#RFASA      ;SET REFRESH START ADDRESS
322      322 001546          ,          WTPS     R4,#RFAIL      ;AND REFRESH LIMIT ADDRESS
323      323 001576          ,          CLR     NUFRAM      ;RESET NEW FRAME FLAG
324      324 001624 005067 176626          ,
325
326      ; RESTART REFRESH CYCLE
327
328      4$:      RWPS      BIS     #20000,#RFSR      ;RESTART REFRESH
329      329 001630
330
331      ; ACKNOWLEDGE RTC INTERRUPT
332
333      5$:      RWPS      BIS     #1,#RTCREQ      ;ACKNOWLEDGE INTERRUPT
334      334 001660
335
336      ; RESTORE SYSTEM TO STATE PRIOR TO INTERRUPT
337
338      NDONE:    TST     IOST        ;DIO READY?
339      BPL     NDONE          ;IF PL NO...WAIT FOR IT
340      MOV     (SP)+,DIOPSA      ;RESTORE DIOPSA
341      MOV     (SP)+,R4          ;RESTORE R4
342      MOV     (SP)+,R5          ;RESTORE R5
343
344      NOINT:    RTI              ;RETURN FROM INTERRUPT
345

```

APPENDIX A: SAMPLE PROGRAM
DATA TO INPUT?

MACRO M11

11-JAN-77 08:57 PAGE 7

347				.SBTTL DATA TO INPUT?	
348					
349	001730	016700	177570	PL00P: MOV SWITCH,R0	;GET SWITCH VALUE
350	001734	001775		BEO PL00P	;AND LOOP IF NOTHING IS SET
351				::: CHECK SWITCH 15	
352				:::	
353				:::	
354	001736	032700	100000	BIT #100000,R0	;SW 15 SET?
355	001742	001402		BEO SW14	;BRANCH IF NOT
356	001744	000167	176524	JMP START	;RESTART
357				::: CHECK SWITCH 14	
358				:::	
359				:::	
360	001750	032700	040000	SW14: BIT #40000,R0	;SW 14 SET?
361	001754	001403		BEO SW13	;BRANCH IF NOT
362	001756	166767	176476 176266	SUB INCVAL,TX	;TRANSLATE -X
363				::: CHECK SWITCH 13	
364				:::	
365				:::	
366	001764	032700	020000	SW13: BIT #20000,R0	;SW 13 SET?
367	001770	001403		BEO SW12	;BRANCH IF NOT
368	001772	166767	176462 176254	SUB INCVAL,TY	;TRANSLATE -Y
369				::: CHECK SWITCH 12	
370				:::	
371				:::	
372	002000	032700	010000	SW12: BIT #10000,R0	;SW 12 SET?
373	002004	001403		BEO SW11	;BRANCH IF NOT
374	002006	166767	176446 176242	SUB INCVAL,TZ	;TRANSLATE -Z
375				::: CHECK SWITCH 11	
376				:::	
377				:::	
378	002014	032700	004000	SW11: BIT #4000,R0	;SW 11 SET?
379	002020	001403		BEO SW10	;BRANCH IF NOT
380	002022	066767	176432 176222	ADD INCVAL,TX	;TRANSLATE +X
381				::: CHECK SWITCH 10	
382				:::	
383				:::	
384	002030	032700	002000	SW10: BIT #2000,R0	;SW 10 SET?
385	002034	001403		BEO SW09	;BRANCH IF NOT
386	002036	066767	176416 176210	ADD INCVAL,TY	;TRANSLATE +Y
387				::: CHECK SWITCH 9	
388				:::	
389				:::	
390	002044	032700	001000	SW09: BIT #1000,R0	;SW 9 SET?
391	002050	001403		BEO DSPLAY	;BRANCH IF NOT
392	002052	066767	176402 176176	ADD INCVAL,TZ	;TRANSLATE +Z

```
395
396
397 002060
398
399
400
401 002060 012700 074377
402
403 002064 012701 177776
404 002070 012702 000152
405
406
407
408 002074 005767 176356
409 002100 001375
410
411 002102 004767 000254
412
413
414
415 002106 012700 140377
416 002112 012701 177774
417 002116 012702 000262
418 002122 004767 000234
419
420
421
422 002126 012700 050377
423
424 002132 012701 177776
425 002136 012702 000272
426 002142 004767 000214
427
428
429
430 002146 012700 074374
431
432 002152 012701 177770
433 002156 012702 000276
434 002162 004767 000174
435
436
437
438 002166 012700 074377
439
440 002172 012701 177776
441 002176 012702 000156
442 002202 004767 000154
443
444
445
446 002206 012700 022360
447 002212 005001
448
449 002214 004767 000142
450
451
```

```
.SBTTL PICTURE DISPLAY
DSPLAY:
;
; SET BLINK MODE ON
;
MOV #DRAW2D+34377,R0 ;SET THE 2DDRAW COMMAND
;FSM1=0, FSM2=7
MOV #-2.,R1 ;2 WORDS TO BE OUTPUT
MOV #BLNKON,R2 ;SET THE BLINK BASE ADDRESS
;
; WAIT FOR THE NEW FRAME FLAG TO BE CLEARED BEFORE OUTPUT
;
DSP010: TST NUFRAM ;IS IT =0 YET?
BNE DSP010 ;BRANCH IF NOT
;
JSR PC,DMAOUT ;NEW FRAME... DMA IT OUT
;
; RESET THE BASE REGISTER
;
MOV #SETBAS,R0 ;SET THE SETBASE COMMAND
MOV #-4.,R1 ;4 WORDS TO BE LOADED
MOV #ZWBASE,R2 ;SET THE BASE ADDRESS
JSR PC,DMAOUT ;AND DMA IT OUT
;
; DO THE SET-POINT FOR THE TEXT
;
MOV #DRAW2D+10377,R0 ;SET THE 2DDRAW COMMAND
;FSM1=0, FSM2=2
MOV #-2.,R1 ;2 WORDS TO BE OUTPUT
MOV #SETPT,R2 ;SET THE DATA ADDRESS
JSR PC,DMAOUT ;AND DMA IT OUT
;
; NOW OUTPUT THE ASCII CHARACTERS
;
MOV #DRAW2D+34374,R0 ;SET THE 2DDRAW COMMAND
;FSM1=0, FSM2=7
MOV #-8.,R1 ;8 WORDS OF CHARACTERS (2/WORD)
MOV #TEXT,R2 ;SET THE TEXT ADDRESS
JSR PC,DMAOUT ;AND DMA IT OUT
;
; SET BLINK MODE OFF
;
MOV #DRAW2D+34377,R0 ;SET THE 2DDRAW COMMAND
;FSM1=0, FSM2=7
MOV #-2.,R1 ;2 WORDS TO BE OUTPUT
MOV #BLNKOF,R2 ;SET THE BLINK OFF BASE ADDRESS
JSR PC,DMAOUT ;AND DMA IT OUT
;
; PUSH THE CURRENT TRANSFORMATION MATRIX
;
MOV #PUSH,R0 ;SET THE PUSH COMMAND
CLR R1 ;A DATALESS COMMAND
;R2 CAN BE ANYTHING
JSR PC,DMAOUT ;AND DMA IT OUT
;
; CONCATENATE THE PERSPECTIVE MATRIX
```

APPENDIX A: SAMPLE PROGRAM
PICTURE DISPLAY

MACRO M11

11-JAN-77 08:57 PAGE 9-1

```

452
453 002220 012700 026000      ;
454 002224 012701 177760      MOV      #MATCON,R0      ;SET THE MATCON COMMAND
455 002230 012702 000162      MOV      #-16.,R1      ;16 WORDS WILL BE OUTPUT
456 002234 004767 000122      MOV      #PERMAT,R2      ;SET THE MATRIX BASE ADDRESS
457                                JSR      PC,DMAOUT      ;AND DMA IT OUT
458
459      ;
460      ;CONCATENATE THE TRANSLATION MATRIX
461      ;
462      MOV      #MATCON,R0      ;SET THE MATCON COMMAND
463      MOV      #-16.,R1      ;16 WORDS WILL BE OUTPUT
464      MOV      #TRNMAT,R2      ;SET THE MATRIX BASE ADDRESS
465      JSR      PC,DMAOUT      ;AND DMA IT OUT
466
467      ;
468      ;RESET THE BASE REGISTER
469      ;
470      MOV      #SETBAS,R0      ;SET THE SETBASE COMMAND
471      MOV      #-4,R1      ;4 WORDS TO BE LOADED
472      MOV      #ZWBASE,R2      ;SET THE BASE ADDRESS
473      JSR      PC,DMAOUT      ;AND DMA IT OUT
474
475      ;
476      ;NOW DRAW THE CUBE (IN 2 INSTALLMENTS)
477      ;
478      MOV      #DRAW3D+21366,R0 ;SET THE BASIC 3D DRAW COMMAND
479      MOV      #-30.,R1      ;FSM1=2, FSM2=4
480      MOV      #CDA11,R2      ;30 WORDS TO BE OUTPUT
481      JSR      PC,DMAOUT      ;SET THE DATA BASE ADDRESS
482      ;AND DMA IT OUT
483
484      ;
485      MOV      #DRAW3D+20372,R0 ;SET THE BASIC 3D DRAW COMMAND
486      MOV      #-18.,R1      ;FSM1=0, FSM2=4
487      MOV      #CDA12,R2      ;18 WORDS TO BE OUTPUT
488      JSR      PC,DMAOUT      ;SET THE DATA BASE ADDRESS
489      ;AND DMA IT OUT
490
491      ;
492      ;INITIATE THE DISPLAY OF THE DATA (SET THE NEW FRAME FLAG)
493      ;
494      INC      NUFRAM
495
496      ;
497      ;POP THE ORIGINAL TRANSFORMATION MATRIX BACK
498      ;
499      MOV      #POP,R0      ;SET THE POP COMMAND
500      CLR      R1      ;A DATALESS COMMAND
501      JSR      PC,DMAOUT      ;AND DMA IT OUT
502
503      ;
504      ;RESTART THE DISPLAY LOOP
505      ;
506      JMP      PLOOP      ;AND CONTINUE

```

```

500          .SBTTL  DMA OUTPUT ROUTINE
501
502          ;
503          ; THIS ROUTINE DOES ALL OF THE DMA OUTPUT FOR THIS PROGRAM.
504          ; EXPECTED PARAMETERS ARE:
505          ; R0=RSR COMMAND
506          ; R1=DMA WORD COUNT FOR DRWC REGISTER
507          ; R2=DMA BASE ADDRESS FOR DRBA REGISTER
508          ;
509 002362    032767    000200    167670  DMAOUT: BIT      #200,IOST      ;IS THE DMA READY?
510 002370    001774                                DMAOUT      ;IF EQ NO...WAIT FOR IT
511
512 002372                                1$:
513 002372                                RWPS      TST      #MSR      ;PPDONE?
514 002420    100364                                BPL      1$      ;IF PL NO...WAIT FOR IT
515
516 002422                                WTPS      R0,#MPIP      ;OUTPUT THE RSR
517 002450    010167    167664                                MOV      R1,DMAWC      ;SET THE DMA WORD COUNT
518 002454    002005                                BGE      2$      ;IF GE NOTHING TO DO
519 002456    010267    167666                                MOV      R2,DMA8A      ;SET THE DMA BASE ADDRESS
520 002462    052767    000001    167670                                BIS      #1,IOST      ;INITIATE THE TRANSFER
521 002470    000207                                RTS      PC      ;AND RETURN
522
523          .LIST  TIM
524          .END  START
000474'

```


APPENDIX A: SAMPLE PROGRAM
SYMBOL TABLE

MACRO M11

11-JAN-77 08:57 PAGE 10-1

BEGIN	000000R	MAOA	=	177751	RTCIE	=	177761
BLNKOF	000156R	MAOL	=	177750	RTCINT	=	001300R
BLNKUN	000152R	MAPREG	=	000000R	RTCREQ	=	177760
BUFPTR	000462R	MATCON	=	026000	RTCSR	=	177745
CDAT1	000316R	MMSR	=	177754	SEIBAS	=	140377
CDAT2	000412R	MPIP	=	177777	SEIPT	=	000272R
DIOPSA	= 167662	MSR	=	177753	START	=	000474R
DMABA	= 167666	NDONE	=	001710R	SWITCH	=	177570
DMAOUT	002362R	NOINT	=	001726R	SW09	=	002044R
DMAPSA	= 177747	NUFRAM	=	000456R	SW10	=	002030R
DMAWC	= 167664	PERMAT	=	000162R	SW11	=	002014R
DRAW2D	= 040000	PGINIT	=	000142R	SW12	=	002000R
DRAW3D	= 100000	PLOOP	=	001730R	SW13	=	001764R
DSPLAY	002060R	POP	=	024360	SW14	=	001750R
DSP010	002074R	PSDATA	=	167660	TEXT	=	000276R
INCVAL	000460R	PUSH	=	022360	TRNMAT	=	000222R
IOST	= 167670	RBTB	=	000466R	TX	=	000252R
LGPIP	= 177775	RFAIL	=	177736	TY	=	000254R
LOAD	= 010000	RFASA	=	177735	TZ	=	000256R
LOWVEC	= 000340	RFSR	=	177737	ZWBASE	=	000262R
MAIA	= 177752	RTCCNT	=	177744			

. ABS. 000000 000
002472 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 736 WORDS (3 PAGES)
DYNAMIC MEMORY: 3188 WORDS (12 PAGES)
ELAPSED TIME: 00:00:23
DK2: SAMPLE/CR=DK2: SAMPLE

SYMBOL CROSS REFERENCE

CREF

SYMBOL	VALUE	REFERENCES				
BEGIN	000000 R	#4-132	5-211			
BLNKOF	000156 R	#4-162	9-441			
BLNKON	000152 R	#4-161	9-404			
BUFPTR	000462 R	#4-202	*5-250	6-317	*6-321	
CDAT1	000316 R	#4-192	9-477			
CDAT2	000412 R	#4-197	9-483			
DIOPSA	= 167662	#3-94	*5-225	*5-232	*5-240	*5-241
		*5-242	*5-252	*5-253	*5-259	*5-275
		*5-276	*5-277	6-299	*6-304	*6-314
		*6-316	*6-318	*6-319	*6-322	*6-323
		*6-329	*6-334	*6-340	*10-513	*10-516
DMABA	= 167666	#3-96	*10-519			
DMAOUT	002362 R	5-237	9-411	9-418	9-426	9-434
		9-442	9-449	9-456	9-463	9-470
		9-478	9-484	9-494	#10-509	10-510
DMAPSA	= 177747	#3-107	5-225			
DMAWC	= 167664	#3-95	*10-517			
DRAW2D	= 040000	#3-125	9-401	9-422	9-430	9-438
DRAW3D	= 100000	#3-126	9-474	9-480		
DSPLAY	002060 R	5-290	7-391	#9-397		
DSP010	002074 R	#9-408	9-409			
INCVAL	000460 R	#4-201	7-362	7-368	7-374	7-380
		7-386	7-392			
IOST	= 167670	#3-97	*5-215	5-225	5-225	5-232
		5-232	5-240	5-240	5-241	5-241
		5-242	5-242	5-252	5-252	5-253
		5-253	5-259	5-259	5-275	5-275
		5-276	5-276	5-277	5-277	*5-279
		6-297	6-300	6-304	6-304	6-310
		6-314	6-314	6-316	6-316	6-318
		6-318	6-319	6-319	6-322	6-322
		6-323	6-323	6-329	6-329	6-334
		6-334	6-338	10-509	10-513	10-513
		10-516	10-516	*10-520		
LGPIP	= 177775	#3-118	5-259			
LOAD	= 010000	#3-121	5-234			
LOWVEC	= 000340	#3-91	5-265			
MAIA	= 177752	#3-111				
MAQA	= 177751	#3-110	5-240	6-316	6-318	
MAOL	= 177750	#3-109	3-110	3-111	3-112	3-113
		5-241	6-319			
MAPREG	000000 R	#4-133	5-236			
MATCON	= 026000	#3-124	9-453	9-460		
MMSR	= 177754	#3-113	5-232			
MPIP	= 177777	#3-119	5-225	10-516		
MSR	= 177753	#3-112	5-242	6-314	10-513	
NDONE	001710 R	#6-338	6-339			
NOINT	001726 R	5-268	5-270	5-272	#6-344	
NUFRAM	000456 R	#4-200	*5-283	6-307	*6-324	9-408
		*9-488				
PERMAT	000162 R	#4-166	9-455			
PGINIT	000142 R	#4-156	5-258			

SYMBOL CROSS REFERENCE

CREF

SYMBOL	VALUE	REFERENCES
PLLOOP	001730 R	#7-349 7-350 9-498
POP	= 024360	#3-123 9-492
PSDATA	= 167660	#3-93 3-94 3-95 3-96 3-97
		*5-225 *5-232 *5-240 *5-241 *5-242
		*5-252 *5-253 *5-259 *5-275 *5-276
		*5-277 6-302 6-304 6-314 6-316
		*6-318 *6-319 *6-322 *6-323 *6-329
		*6-334 10-513 *10-516
PUSH	= 022360	#3-122 9-446
RBTB	000466 R	#4-205 *5-246 *5-247 *5-248 *5-249
		6-318 6-319 6-322 6-323
RFAIL	= 177736	#3-101 5-253 5-252 6-322
RFASA	= 177735	#3-100 3-101 3-102 5-252 6-322
RFSR	= 177737	#3-102 6-304 6-329
RTCCNT	= 177744	#3-104 3-105 5-276
RTICIE	= 177761	#3-116 5-275
RTCINT	001300 R	5-266 #6-295
RTCREQ	= 177760	#3-115 3-116 6-334
RTCSR	= 177745	#3-105 5-277
SETBAS	= 140377	#3-127 9-415 9-467
SETPT	000272 R	#4-187 9-425
START	000474 R	#5-211 7-356 10-524
SWITCH	= 177570	#3-98 7-349
SW09	002044 R	7-385 #7-390
SW10	002030 R	7-379 #7-384
SW11	002014 R	7-373 #7-378
SW12	002000 R	7-367 #7-372
SW13	001764 R	7-361 #7-366
SW14	001750 R	7-355 #7-360
IEXT	000276 R	#4-188 9-433
TRNMAT	000222 R	#4-174 9-462
IX	000252 R	#4-177 *5-284 *7-362 *7-380
IY	000254 R	#4-178 *5-285 *7-368 *7-386
IZ	000256 R	#4-179 *5-286 *7-374 *7-392
ZWBASE	000262 R	#4-183 9-417 9-469

SAMPLE CREATED BY MACRO ON 11-JAN-77 AT 08:57 PAGE 3

MACRO CROSS REFERENCE

CREF

MACRO NAME REFERENCES

RDPS	#2-34	6-316			
RWPS	#2-58	#5-232	#5-240	#5-242	#5-277
	#6-304	#6-314	#6-329	#6-334	#10-513
WTPS	#2-46	5-225	5-241	5-252	5-253
	5-259	5-275	5-276	6-318	6-319
	6-322	6-323	10-516		